Analysis in Univalent Type Theory

Auke Bart Booij

A thesis submitted to the
University of Birmingham
for the degree of
DOCTOR OF PHILOSOPHY

School of Computer Science
College of Engineering and Physical Sciences
University of Birmingham
January 2020

ABSTRACT

Some constructive real analysis is developed in univalent type theory (UTT). We develop various types of real numbers, and prove several equivalences between those types. We then study computation with real numbers. It is well known how to compute with real numbers in intensional formalizations of mathematics, where equality of real numbers is specified by an imposed equivalence relation on representations such as Cauchy sequences. However, because in UTT equality of real numbers is captured directly by identity types, we are prevented from making any nontrivial discrete observations of arbitrary real numbers. For instance, there is no function which converts real numbers to decimal expansions, as this would not be continuous. To avoid breaking extensionality, we thus restrict our attention to real numbers that have been equipped with a simple structure called a *locator*. In order to compute, we modify existing constructions in analysis to work with locators, including Riemann integrals, intermediate value theorems and differential equations. Hence many of the proofs involving locators look familiar, showing that the use of locators is not a conceptual burden. We discuss the possibility of implementing the work in proof assistants and present a Haskell prototype.

ACKNOWLEDGEMENTS

This dissertation exists thanks to the detailed, patient and sympathetic supervision of Martín Escardó and Benedikt Ahrens. Thanks to the Birmingham Theory group, and in particular the CARGO category theory reading group for many opportunities to have (often opinionated) discussions on constructive mathematics.

Thanks to Thorsten Altenkirch, Andrej Bauer, Mark Bickford, Ulrik Buchholtz, Achim Jung, Milly Maietti, Peter Schuster, Bas Spitters, Steve Vickers and various anonymous reviewers for their input and inspiring discussions. Thanks to Hajime Ishihara and Tomoaki Hashizaki for their care and company during my stay at the Japan Advanced Institute of Science and Technology. Thanks to Sebastian Faust for welcoming me in his group without question after my move to Darmstadt.

Thanks to my parents, family, and friends for continued support throughout my studies. Kristina, thank you for your love and support.

Thanks to the computing facilities team at the School of Computer Science for providing the most thankless service in IT: fast and stable Wi-Fi.

This project has received funding from the School of Computer Science at the University of Birmingham, and from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 731143.

CONTENTS

1 **Introduction** 1

1.1 Summary of contributions 4

2 **Univalent mathematics** 7

2.1 Case study: propositional logic 8

2.1.1 Formalizing the theorem statement 8

2.1.2 Formalizing the proof 10

2.2 Proofs versus derivations 11

2.3 Case study: every natural is either even or odd 12

2.3.1 Formalizing the theorem statement 13

2.3.2 Formalizing the proof 22

2.4 Case study: surjective maps, images of maps 27

2.4.1 Formalizing the theorem statement 28

2.4.2 Formalizing the proof 36

2.5 Extensionality 36

2.5.1 Function extensionality 37

2.5.2 Contractibility 39

2.5.3 Computation rules for higher-inductive types 41

2.5.4 Univalence 42

2.5.5 Consequences of extensionality 45

2.6 Subtypes and embeddings . . . 46
2.6.1 Lattice-like structure of HProp . . . 50
2.6.2 Quantification over subtypes . . . 52
2.6.3 Identifications in subtypes . . . 53
2.7 Case study: quotient types . . . 54
2.8 Classical principles . . . 56
2.9 Notes . . . 58

3 Fixpoints in dcpos 61
3.1 Dcpos . . . 62
3.2 Fixpoints . . . 64
3.3 Notes . . . 66

4 Real numbers 67
4.1 Algebraic structure of numbers . . . 69
4.2 Rationals . . . 75
4.3 Archimedean property . . . 76
4.4 Cauchy completeness of real numbers . . . 77
4.5 HoTT book reals . . . 81
4.6 Dedekind reals . . . 89
4.7 Notes . . . 92

5 Universal properties of real numbers 95
5.1 Subsets of the Dedekind reals . . . 97
5.1.1 Minimality of the HoTT book reals . . . 98
5.1.2 Euclidean reals and interval objects . . . 100
5.2 Homotopy-initiality of the Euclidean reals . . . 101
5.3 Interval objects . . . 111
5.4 Notes . . . 116

6 Locators 119

6.1 Definition 120

6.2 Terminology for locators 122

6.3 Locators for rationals 122

6.4 The logic of locators 123

6.5 Bounded search 127

6.6 Computing bounds 128

6.7 Locators for algebraic operations 130

6.8 Locators for limits 137

6.9 Calculating digits 138

6.10 Dedekind cuts structure 141

6.11 Notes 145

7 Some constructive analysis with locators 147

7.1 Preliminaries 147

7.2 Integrals 150

7.3 Intermediate value theorems 151

7.4 Notes 153

8 Metric spaces 155

8.1 Basic definitions in pseudometric spaces 156

8.2 Function spaces 159

8.3 Banach fixed point theorem 163

8.4 Picard-Lindelöf 167

8.5 Notes 170

9 Computation in proof assistants 173

9.1 From inference rules to proof assistants 173

9.2 Locators 174

9.3 Metric spaces 176

9.4 Haskell prototype 177

10 Closing remarks **181**

Index **185**

Bibliography **189**

Chapter 1

INTRODUCTION

Traditionally important parts of mathematics can be developed in a constructive setting, as Bishop showed in his self-proclaimed constructivist propaganda [18]. Like most mathematicians before Bishop, including constructive mathematicians such as Brouwer, Bishop worked informally, although he did envision formal systems that allowed for computationally executing the finite routines he described [17, 19].

Martin-Löf [75] proposed a formal system in which Bishop's work was supposed to be formalizable. This formal system and several of its variations are now known as Martin-Löf Type Theory (MLTT). Interpretation of Bishop's mathematics in MLTT has been described in e.g. Coquand and Spiwack [35, Section 3].

It is well known how to compute with real numbers in such an intensional formalization, with equality of real numbers specified by an imposed equivalence relation on representations [20, 36, 77], such as Cauchy sequences, sequences of nested intervals, or streams of digits. It has to be checked explicitly that functions on the representations preserve these equivalence relations. Discrete observations, such as finite decimal approximations, can be made because representations are given, but a different representation of the same real number can result in a different observation, and hence discrete observations are necessarily non-extensional.

Interpretation of the so-called *identity types* remained an issue in MLTT. Such identity types represent the logical notion of equality, and so mirroring first-order logic, were not given any structure beyond being thought of as containing at most one element, despite this seemingly not being provable in MLTT. The genie was let out of the bottle by the development of a model of MLTT in groupoids, where identity types can contain more than one element [53].

The potential non-uniqueness of identifications was exploited in an explanation of identity types originating from homotopy theory [46, 11, 59]. The development of homotopy theory in terms of type theory led to the *univalence axiom* which reinforces this explanation and makes non-uniqueness of identifications provable.

The resulting *univalent type theory* (UTT), which we discuss in Chapter 2, allows to formalize mathematics with the identity types in a much more central role than in plain MLTT. For example, we can define types of real numbers whose identity types directly capture the intended equality of real numbers, rather than this equality being captured by a defined equivalence relation. See also Chapter 4 for some types of reals.

Certain types in UTT can be characterized by *universal properties* in the sense of category theory [71, 9], again thanks to the well-behavedness of identity types in UTT. For the particular case of the real numbers, we are reminded of the characterization of reals in terms of *interval objects* [43], and so we compare some notions of reals using this terminology in Chapter 5.

All functions automatically preserve identifications of real numbers for type-theoretic reasons. As a consequence, nontrivial discrete observations of arbitrary real numbers would violate continuity principles. So, as a drawback of this automatic preservation, we cannot in general tolerate a construction of a decimal expansion for any real number, as this would not be continuous. This kind of problem is already identified in Hofmann [52, Section 5.1.7.1] for an extensional type theory. Hofmann solves this by making discrete observations of real numbers using an extensionality-breaking *choice operator*, which does not give rise to a *function*.

Altenkirch, Danielsson, and Kraus [3] and Gilbert [48] make observations of Cauchy reals (Section 4.4) and HoTT book reals (Section 4.5) using a *delay monad* defined as a quotient inductive-inductive type. Their constructions depend on a specific definition of the type of reals, and so it is not clear whether such an approach works more generally, for example with the Dedekind reals (Section 4.6).

To avoid breaking extensionality in UTT, the central idea of Chapters 6–8 is to restrict our attention to real numbers that have been equipped with a simple structure called a *locator*.

Such a locator is a strengthening of the locatedness property of Dedekind cuts, although we work with an arbitrary type of reals. While the locatedness of a real number x says that for rational numbers $q < r$ we have the property $q < x$ or $x < r$, a locator produces a specific selection of one of $q < x$ and $x < r$. In particular, the same real number can have different locators, and it is in this sense that locators are structure rather than property.

In a constructive setting such as ours, it cannot be proved that all real numbers have locators, and we prove that the ones that do are the ones that have Cauchy representations in Section 6.9. However, working with locators rather than Cauchy representations gives a development which is closer to that of traditional real analysis. For example, we can prove that if x has a locator, then so does e^x, and this allows to compute e^x when working constructively, so that we say that the exponential function *lifts to locators.* As another example, if f is given a modulus of continuity and lifts to locators, then $\int_0^1 f(x)\,\mathrm{d}x$ has a locator and we can compute the integral in this way.

Thus the difference between locatedness and locators is that one is property and the other is structure. Plain Martin-Löf type theory is not enough to capture this distinction because, for example, it allows to define the notion of locator as structure but not the notion of locatedness as property, and therefore it does not allow to define the notion of real numbers we have in mind, whose identity type should capture directly the intended notion of equality of real numbers. For us, most of the time it is enough to work in the fragment of UTT consisting of MLTT with propositional truncation (introduced in Section 2.4.1), and propositional extensionality and function extensionality (introduced in Section 2.5). However, there a few exceptions that use the more general extensionality principle of univalence, such as

1. our definition of the HoTT book reals, and in particular Theorem 4.5.13 (Cauchy completeness by obtaining an induction principle from a universal property), and

2. Lemmas 2.6.3 and 2.8.1.

The need for univalence would also arise when considering types of sets with structure such as the type of metric spaces or the type of Banach spaces for the purposes of functional analysis.

We believe that our constructions can also be carried out in other constructive foundations such as CZF, the internal language of an elementary topos with a natural numbers object, or Heyting arithmetic of finite types. Our results being phrased in UTT can be seen as a pragmatic choice, as it is a constructive system with sufficient extensionality, which admits, at least in principle, applications in proof assistants such as cubical Agda allowing for computation using the techniques in this work.

1.1 Summary of contributions

We have chosen to introduce UTT by case studies in Chapter 2, rather than building the theory from the ground up, which seems to be more common. The development otherwise essentially follows The Univalent Foundations Program [91]. We have done our best to avoid talking about dependent identifications as we feel these do not add pedagogical value.

The development of dcpos in Chapter 3 is a translation of existing work into type theory, where we have paid special attention to universe levels, which were not an issue in the impredicative settings used previously. We have minutely rephrased the proof of Pataraia's fixed point theorem by adding Corollary 3.2.2 as an intermediary claim.

The definition of ordered fields in Chapter 4 is a simplification of the one in The Univalent Foundations Program [91, Definition 11.2.7]. We have elaborated on the inclusion of the rationals into ordered fields. The presentation of the HoTT book reals as an initial *Cauchy algebra*, rather than by its inference rules as a HIIT, is new. The Dedekind reals are presented by overloading the inequality relation $<$, so that we can conveniently write $q < x$ to mean that the rational q is in the left cut of x.

We characterize the HoTT book reals as a certain initial subset of the Dedekind reals in Section 5.1, which answers a question raised in The Univalent Foundations Program [91, Chapter 11] positively. The homotopy-initiality of the Euclidean reals in Theorem 5.2.1, and an equivalent definition of interval objects in Theorem 5.3.4, are new to the best of our knowledge.

Chapter 6 gives the basic theory of the new notion of *locators*, and consequently all results that mention locators are new, although many results can be said to have equivalents in terms of intensional representations of reals. For instance, while we can compute a rational lower bound $q < x$ from a *locator* for x in Lemma 6.6.1, it was already known how to compute such a lower bound when x is presented as a Cauchy sequence of rationals, or as a sequence of nested intervals. We consider Theorems 6.9.7 and 6.10.3 to be the omnibus theorems of locators.

Many of the *proofs* in Chapter 6, including proofs of claims about locators, are very similar to existing proofs in constructive analysis: the reader is invited to rather pay attention to the claims themselves, and in particular to our focus on the distinction between when such claims are made as *property*, and when they are made as *structure*, after these terms have been defined in Chapter 2.

We have developed constructive analysis with locators in Chapter 7 by introducing the notion of *lifting to locators*. We rephrase the proof of a known approximate intermediate value theorem and of the computation of integrals to additionally work with locators, so that in the same spirit as that of the previous chapter, we may say that it is not the proof but the statement that is new. Our exact intermediate value theorem, namely Theorem 7.3.5, improves on existing results stated in terms of representations by assuming local nonconstancy as *property* rather than as *structure*.

Chapter 8, too, takes known results in constructive analysis and upgrades them to locators, so that it is our focus on property versus structure in the claims, rather than any particular proof, that is new.

The main contribution of the thesis is to redevelop constructive analysis in a way that is closer to classical analysis by working with locators, so that we can focus on real numbers rather than their representations. Additionally, this allows to compute with the proofs using proof assistants, and this prospect is discussed in Chapter 9.

Chapter 2

UNIVALENT MATHEMATICS

We discuss the formalization of mathematics in univalent type theory (UTT), which may be described as dependent type theory with additional inference rules. In such type theories, we reason mathematically by constructing elements of types, with the constraint that we only talk about *typed* elements. The notation $\Gamma \vdash M : A$ asserts that M is a *term* whose *type* is A, optionally using (typed) bound variables, and free variables specified by the *context* Γ. If the type-theoretic context is clear from the linguistic context, that is, whenever the types of the free variables of M and A are implicit, we simply write $M : A$. We write $M, N : A$ for $M : A$ and $N : A$. Concretely, typing means that integers are not also rationals—at best we relate the integer 5 and the rational 5 via a choice of inclusion $\mathbb{Z} \to \mathbb{Q}$.

In a simple type theory such as simply typed lambda calculus, the elements live either in an atomic type T or a function type $\sigma \to \tau$ where σ and τ are types. In other words, it is a type theory of higher-order functions. For example, we have the term $\lambda(f : T \to T).\lambda(x : T).f(f(x))$ of type $(T \to T) \to (T \to T)$, which takes a function f and applies it twice to an input x.

Dependent type theory is more expressive in the sense that it has *dependent types*, which we'll describe in detail below, as well as certain basic types and type formers. Such a type theory is sufficiently expressive to encode a wide range of mathematics.

Univalent type theory adds certain extensionality principles as well as additional type formers to dependent type theory. These additions allow us to give *identity types*, which are already present in dependent type theory, a more central role.

At times we will relate type-theoretic concepts to set theory and Bishop mathematics. We emphasize that the purpose of this is to show that certain type-theoretic concepts and constructions are unsurprising, and expressly *not* to motivate the usage of type theory as a foundation. The only requirement of alternative foundations of mathematics is that they allow the accurate interpretation of informal mathematics: informal definitions and constructions can be made precise, and it should be possible to use informal proofs as strategies for building formal proofs of formalized theorems, all following the inference rules of, in this case, univalent type theory.

We explain how univalent type theory works by example, by describing how to formalize some theorems and their proofs.

2.1 Case study: propositional logic

Theorem. *If P or Q, then $\neg P$ implies Q.*

Note that in constructive mathematics, negation of P is defined by the statement that truth of P implies a contradiction.

Proof. Assume P or Q.

If P, then further assume $\neg P$, which is a contradiction, and hence we may conclude Q.

If Q, assume $\neg P$ and conclude Q. □

2.1.1 Formalizing the theorem statement

Claims such as P and Q and the theorem statement are formalized by types, and proofs of the theorem are formalized by elements of the corresponding type.

Discussion 2.1.1. As is usual in logic, by a *term* we mean a well-typed expression in the language of UTT. This contrasts with *elements* which are the more general constituents of a type. So when we have assumed to have $n : \mathbb{N}$, that means we have taken an arbitrary *element* n of the type $\mathbb{N}$ of natural numbers. The *expression* n is then a term. An expression such as $\mathrm{succ}(0)$ would be a specific *term*, also of type $\mathbb{N}$. When making some mathematical claim *in* the language of UTT, we can't take an arbitrary *term*, since this is a concept *of* the language of UTT. So the distinction between *term* and *element* is the same as that between *numerals* and *numbers*.

The claim is an implication from "P or Q" to "$\neg P$ implies Q". Implications are formalized as function types, and we have the empty type $\mathbf{0}$ which formalizes the contradiction in $\neg P$. So after formalizing the disjunction, we can formalize the theorem statement as

$$\text{“}P \text{ or } Q\text{”} \to (\neg P \to Q).$$

Discussion 2.1.2. Now we discuss the formalization of the disjunction "P or Q". In univalent type theory, there are two ways to disjoin two types X and Y, where the difference between these two choices is central to our work:

1. We can form the coproduct $X + Y$ whose elements are either an element of X on the left, or an element of Y on the right. One can think of $X + X$ as containing twice as many elements as X.

2. We can form the propositional disjunction $X \vee Y$, which contains at most one element, which represents that one of X or Y is non-empty. It is not possible in general to obtain a specific element of X or Y from a proof of $X \vee Y$.

The claim that $X \vee Y$ contains at most one element can be made precise type-theoretically using identity types, which formalize when two elements of a fixed type are to be considered equal. We discus identity types below.

The distinction between a choice such as a coproduct and a propositional truth value such as the disjunction $X \vee Y$ is important. For instance, if certain Brouwerian continuity principles are stated incorrectly, they are false, rather than independent of the type theory[44].

To keep the formalization of the proof simple, we choose the former option as formalization of the theorem:

$$P + Q \to (\neg P \to Q).$$

2.1.2 Formalizing the proof

We have to find a term whose type is

$$P + Q \to (\neg P \to Q).$$

In other words, we have to define a function whose input is an element of type $P + Q$, and whose output is an element of type $\neg P \to Q$.

The specification of coproduct types gives us a *recursion principle* which tells us that we may define a function $X + Y \to Z$ by pattern matching: we have to supply functions $f : X \to Z$ and $g : Y \to Z$, where the first corresponds to the case that the input element of type $X + Y$ is of the form $\mathsf{inl}(x)$ for some $x : X$, and similarly for the second. We then obtain the function as:

$$\mathsf{rec}_{X+Y}(Z, f, g) : X + Y \to Z$$

In our case, we have to define functions $P \to (\neg P \to Q)$ and $Q \to (\neg P \to Q)$.

"$P \to (\neg P \to Q)$": The informal statement that we "assume P" corresponds to λ-abstraction: we take an element $p : P$, and output an element of type $\neg P \to Q$. To do that, we take an element $np : \neg P$, and output an element of type Q. Hence, we seek to implement *ex falso quodlibet* $\mathrm{efq}_Q : \mathbf{0} \to Q$. We appeal to the recursion principle for $\mathbf{0}$ which tells us that we may define a function $\mathbf{0} \to Q$ by pattern matching on nothing: we only have to specify our output type, but we do not have to supply any functions or input data. Hence we can define efq_Q as:

$$\mathrm{efq}_Q :\equiv \mathsf{rec}_{\mathbf{0}}(Q) : \mathbf{0} \to Q.$$

Having defined this, we can define an element of type $P \to (\neg P \to Q)$ as:

$$\lambda(p : P).\lambda(np : \neg P).\, \mathrm{efq}_Q(np(p)) : P \to (\neg P \to Q).$$

"$Q \to (\neg P \to Q)$": Having done the above, this case is straightforward, using two lambda

$$\dfrac{\dfrac{\dfrac{\dfrac{}{P, \neg P \vdash Q}}{P \vdash \neg P \Rightarrow Q} \qquad \dfrac{\dfrac{}{Q, \neg P \vdash Q}}{Q \vdash \neg P \Rightarrow Q}}{P \vee Q \vdash \neg P \Rightarrow Q}}{\vdash (P \vee Q) \Rightarrow (\neg P \Rightarrow Q)}$$

Figure 2.1: An example of a propositional derivation in natural deduction

abstractions:

$$\lambda(q : Q).\lambda(np : \neg P).q : Q \to (\neg P \to Q).$$

Finally, the full formalized proof can be given as:

$$\begin{aligned}&\mathsf{rec}_{P+Q}((\neg P \to Q), \lambda(p : P).\lambda(np : \neg P).\,\mathrm{efq}_Q(np(p)), \lambda(q : Q).\lambda(np : \neg P).q)\\ &\quad : P + Q \to (\neg P \to Q)).\end{aligned}$$

2.2 Proofs versus derivations

The example of Section 2.1 may be shown in natural deduction as in Figure 2.1.

The proof theory of univalent type theory can be seen as a variation of natural deduction: the valid terms are described by a proof theory. A full derivation of the proof term in Section 2.1.2 may be provided in terms of the inference rules of univalent type theory, which are presented e.g. in The Univalent Foundations Program [91, Appendix A.2].

For illustrative purposes, we give one such derivation of a term now. Whenever A is a type, that is, whenever we have been able to define an element A of a universe $\mathcal{U}$ of types, we can define the identity function on A. Read logically, this expresses that for any claim A, the truth of A implies the truth of A. This can be shown by implementing the identity function $\mathrm{id}_A \equiv \lambda(x : A).x$ on A, for example as in Figure 2.2.

We do not prove theorems using complete such derivation trees, because the derivation tree is already encoded by the conclusion. Proof assistants, discussed in Section 9.1, can be used to perform this reconstruction algorithmically. For example, in Figure 2.2, the outermost term constructor is λ-abstraction, and hence the inference rule used to reached the conclusion must

$$\dfrac{\dfrac{\dfrac{\Gamma \vdash A : \mathcal{U}}{\Gamma, x : A \ \mathsf{ctx}}\ \mathsf{ctx}\text{-}\textsc{ext}}{\Gamma, x : A \vdash x : A}\ \mathsf{Vble}}{\Gamma \vdash \lambda(x : A).x : A \to A}\ \Pi\text{-}\textsc{intro}$$

Figure 2.2: The identity function inferred in type theory

have been Π-intro. The type checking mechanism in proof assistants such as Coq and Agda can be thought of as reconstructing these derivation trees. Hence, when doing mathematics inside type theory, we only need to consider the final typing judgments.

2.3 Case study: every natural is either even or odd

Theorem. *Let n be a natural number. Either n is even or n is odd.*

Proof. By induction, it suffices to show that 0 is either even or odd, and that, whenever n is either even or odd, then so is $n + 1$.

0 is even because $0 = 2 \cdot 0$.

Assume n is either even or odd. We need to show that $n + 1$ is either even or odd. If n were even, that is, if $n = 2k$, then $n + 1$ is odd. If n were odd, that is, if $n = 2k + 1$, then $n + 1$ is odd because $n + 1 = (2k + 1) + 1 = 2(k + 1)$. □

In essence, we can formalize this in UTT by closely following the above. First, the theorem statement gets formalized as a type

$$(\Pi n : \mathbb{N})\ \mathrm{isEven}(n) + \mathrm{isOdd}(n)$$

where isEven and isOdd, respectively, are type families expressing that n is respectively even or odd. Then we prove the theorem by finding an element of this type, which we can do using the induction principle $\mathsf{ind}_{\mathbb{N}}$ of the natural numbers. The case $n = 0$ can be shown by constructing an element of the type $\mathrm{isEven}(0) + \mathrm{isOdd}(0)$, and the induction case is shown using an induction principle $\mathsf{ind}_{=_{\mathbb{N}}}$ on identity types.

$$\frac{\Gamma\ \mathsf{ctx}}{\Gamma \vdash \mathbb{N} : \mathcal{U}_i}\ \mathbb{N}\text{-}\textsc{form} \qquad \frac{\Gamma\ \mathsf{ctx}}{\Gamma \vdash 0 : \mathbb{N}}\ \mathbb{N}\text{-}\textsc{intro}_1 \qquad \frac{\Gamma \vdash n : \mathbb{N}}{\Gamma \vdash \mathsf{succ}(n) : \mathbb{N}}\ \mathbb{N}\text{-}\textsc{intro}_2$$

Figure 2.3: Formation and introduction rules governing $\mathbb{N}$ (as presented in The Univalent Foundations Program [91, Appendix A.2]). Γ and n are arbitrary expressions.

We now discuss this in detail. Note that the majority of the text is an explanation of type theory in general, rather than representing an intrinsic part of the formalization of the above theorem.

2.3.1 Formalizing the theorem statement

Types play the role of both logical claims, and of collections of elements, that is, of mathematical constructions. In contrast, in traditional set-based mathematics, the constructions are carried out in sets, but the logic is formalized by a different language such as first-order logic.

In any case we represent the theorem statement by some type. The theorem makes a claim about an arbitrary natural number n, namely that n is either even or odd. Three of the inference rules governing $\mathbb{N}$, displayed in Figure 2.3, say that we can form the type $\mathbb{N}$, and that it has $0 : \mathbb{N}$, and that whenever $n : \mathbb{N}$, we can form $\mathsf{succ}(n) : \mathbb{N}$. In other words, the inference rules specify that we can form the type $\mathbb{N}$, and that it has two *generators*, namely 0 and succ, of appropriate arities. There are a couple more inference rules governing the naturals, which we will get to later in Section 2.3.2. The natural numbers can also be presented as a W-type [10].

Inference rules

We look at these rules in a bit more detail. The assumption Γ ctx in the formation rule says that Γ is a well-formed *context* of variable declarations—intuitively, an ordered list of typed variables, where the types of later variables may be written in terms of earlier ones. The conclusion $\mathbb{N} : \mathcal{U}_i$ says that $\mathbb{N}$ is an element of the ith *universe* of types. The reason for the index i is that we wish to think of a universe $\mathcal{U}$ of types *itself* as a type, but saying that $\mathcal{U} : \mathcal{U}$ would lead to an inconsistency, for example via Girard's paradox [49]. So instead we have a hierarchy $\mathcal{U}_0 : \mathcal{U}_1 : \mathcal{U}_2 : \ldots$ of universes indexed by numerals (i.e. natural numbers of the metatheory), and we write $\mathcal{U}$ for one such $\mathcal{U}_i$, leaving the choice of i undecided or implicit.

Remark 2.3.1. The Coq proof assistant (see also Chapter 9) pretends to have a universe $\mathcal{U} : \mathcal{U}$ which is an element of itself:

```
$ coqtop
Welcome to Coq 8.8.2 (October 2018)


Coq < Check Type.
Type
     : Type
```

When Coq is asked to check the type of `Type`, it returns that `Type` is an element of itself. However, in reality Coq is hiding the universe levels, and we can ask Coq to show them:

```
$ coqtop
Welcome to Coq 8.8.2 (October 2018)


Coq < Set Printing Universes.


Coq < Check Type.
Type@{Top.1}
     : Type@{Top.1+1}
(* {Top.1} |=  *)
```

Now, Coq reports that the universe with a variable index `Top.1` is an element of the universe with index `Top.1+1`, as expected.

Remark 2.3.2. We present a type theory in which there is a context judgment denoted ctx, a term typing judgment denoted $(-) \vdash (-) : (-)$, and two corresponding judgmental

$$\frac{\Gamma \vdash A : \mathcal{U}_i \qquad \Gamma, x : A \vdash B : \mathcal{U}_i}{\Gamma \vdash (\Pi x : A) B : \mathcal{U}_i}\ \Pi\text{-FORM} \qquad\qquad \frac{\Gamma, x : A \vdash b : B}{\Gamma \vdash \lambda(x{:}A).\, b : (\Pi x : A) B}\ \Pi\text{-INTRO}$$

Figure 2.4: Formation and introduction rules governing Π (based on the presentation in The Univalent Foundations Program [91, Appendix A.2]). Γ, A, B and b are arbitrary expressions.

equality judgments (discussed in Section 2.3.2). Another common way to present type theory adds a type judgment that expresses something is a well-formed type in some context, and a corresponding judgmental equality judgment. For example, ℕ-FORM would be presented as:

$$\frac{\Gamma \text{ ctx}}{\Gamma \vdash \mathbb{N} \text{ type}}\ \mathbb{N}\text{-FORM'}$$

The advantage of this approach is that it does not require a hierarchy of universes. However, it requires additional rules that let us see elements of a universe $\mathcal{U}$ as types and vice versa: the elements of $\mathcal{U}$ are considered *codes of types* which may be reified into actual types. This alternative approach is known as universes à la Tarski, whereas we present universes à la Russell.

The statement of these rules in terms of a context Γ is especially important for dependent types, and the contexts also play a role in the induction principle and computation rules for ℕ. We discuss both topics below.

Universal quantification

As we are making a claim about any natural number n, we have to generate a type for every natural number. So, writing isEvenOrOdd(n) for the claim *about* a given n, we require a *type family*, that is, a function

$$\text{isEvenOrOdd} : \mathbb{N} \to \mathcal{U}$$

from the natural numbers to a certain collection $\mathcal{U}$ of types—we formalize functions in a second. Generally speaking, we wish to minimize the number of universes involved in our development, and in this case we are able to choose the first universe $\mathcal{U}_0$ as $\mathcal{U}$. In this work, we will normally abstain from explicitly choose universe levels, and instead pretend to have a universe $\mathcal{U} : \mathcal{U}$.

It is incorrect to say that isEvenOrOdd itself is the formalization of the theorem statement: after all, it is not a type but an element of the type $\mathbb{N} \to \mathcal{U}$. To understand how to formalize the theorem statement, consider what we expect a proof of the theorem to do. It should take as an input a natural number n, and give as an output a proof of $\mathrm{isEvenOrOdd}(n)$, namely a proof that n is either even or odd. In other words, the theorem statement should be a type of *dependent functions* whose input is a natural number n, and whose output is a proof of the claim $\mathrm{isEvenOrOdd}(n)$ *about* that natural number n.

We obtain such a type using the type former for Π-types in Figure 2.4. We have to specify a fixed domain $A : \mathcal{U}$, and for every element of A we have to specify a codomain—in other words, we need a function $B : A \to \mathcal{U}$. Given this data, we can form the type $(\Pi x : A)B(x)$ whose elements are maps that take an element $x : A$ and output an element of $B(x)$. So, after defining isEvenOrOdd, as we will do below, we can formalize the theorem statement as the type

$$(\Pi n : \mathbb{N})\ \mathrm{isEvenOrOdd}(n).$$

We may construct elements of dependent function types using λ-abstraction, as is made precise by the introduction rule for (in)dependent functions as displayed in Figure 2.4. Loosely speaking, it says that if we can specify an output term b in terms of a free variable $x : A$, we may bind the variable x to obtain a function.

The “dependent” in “dependent function” refers to the fact that the output *type*, namely $\mathrm{isEvenOrOdd}(n)$, depends on the input *value* n, in contrast to what one might call “independent functions” of type $X \to Y$ which have a fixed codomain Y which is independent of the input value (of type X).

Choices

In fact, we now define the independent function isEvenOrOdd by λ-abstraction. So it suffices to define $\mathrm{isEvenOrOdd}(n)$ in terms of a given n. $\mathrm{isEvenOrOdd}(n)$ is a logical disjunction of

$$\frac{\Gamma \vdash A : \mathcal{U}_i \qquad \Gamma \vdash B : \mathcal{U}_i}{\Gamma \vdash A + B : \mathcal{U}_i}\ \textsc{+-form}$$

$$\frac{\begin{array}{c}\Gamma \vdash A : \mathcal{U}_i \qquad \Gamma \vdash B : \mathcal{U}_i \\ \Gamma \vdash a : A\end{array}}{\Gamma \vdash \mathsf{inl}(a) : A + B}\ \textsc{+-intro}_1 \qquad\qquad \frac{\begin{array}{c}\Gamma \vdash A : \mathcal{U}_i \qquad \Gamma \vdash B : \mathcal{U}_i \\ \Gamma \vdash b : B\end{array}}{\Gamma \vdash \mathsf{inr}(b) : A + B}\ \textsc{+-intro}_2$$

Figure 2.5: Formation and introduction rules governing + (as presented in The Univalent Foundations Program [91, Appendix A.2]). Γ, A, B, a and b are arbitrary expressions.

the claim that n is even and the claim that n is odd. Recall from Discussion 2.1.2 that there are two ways to disjoin two types X and Y:

1. We can form the coproduct $X + Y$ whose elements are either an element of X on the left, or an element of Y on the right. One can think of $X + X$ as containing twice as many elements as X.

2. We can form the propositional disjunction $X \vee Y$, which contains at most one element, which represents that one of X or Y is non-empty. It is not possible in general to obtain an element of X or Y from a proof of $X \vee Y$.

The difference between these two choices is central to our work. However, in our particular case of n being even or odd, it turns out that the choice does not matter, as "n is even" and "n is odd" are mutually exclusive *propositions*. We discuss propositions in more detail in Section 2.4.1. Hence we formalize $\mathrm{isEvenOrOdd}(n)$ as a coproduct, which is easier to use in most cases. The formation and introduction rules for coproducts are displayed in Figure 2.5.

To formalize when n is even and when n is odd, we wish to say that n is even if there exists a natural k such that $n = 2k$, and that n is odd if there exists a natural k such that $n = 2k + 1 = \mathrm{succ}(2k)$. Given a predicate Q on a type X (formalized as a function $Q : X \to \mathcal{U}$), there are two ways to existentially quantify over the elements of X that satisfy Q, where in our working example X will be $\mathbb{N}$ and Q is the claim about $k : \mathbb{N}$ that $n = 2k$:

1. We can form the *dependent sum*

$$(\Sigma x : X)Q(x)$$

$$\frac{\Gamma \vdash A : \mathcal{U}_i \qquad \Gamma, x : A \vdash B : \mathcal{U}_i}{\Gamma \vdash (\Sigma x : A)B : \mathcal{U}_i}\ \Sigma\text{-FORM}$$

$$\frac{\Gamma, x : A \vdash B : \mathcal{U}_i \qquad \Gamma \vdash a : A \qquad \Gamma \vdash b : B[a/x]}{\Gamma \vdash (a, b) : (\Sigma x : A)B}\ \Sigma\text{-INTRO}$$

Figure 2.6: Formation and introduction rules governing Σ (based on the presentation in The Univalent Foundations Program [91, Appendix A.2]). Γ, *A*, *B*, *a* and *b* are arbitrary expressions.

$$\frac{\Gamma \vdash A : \mathcal{U}_i \qquad \Gamma \vdash a : A \qquad \Gamma \vdash b : A}{\Gamma \vdash a =_A b : \mathcal{U}_i}\ {=}\text{-FORM} \qquad \frac{\Gamma \vdash A : \mathcal{U}_i \qquad \Gamma \vdash a : A}{\Gamma \vdash \mathsf{refl}(a) : a =_A a}\ {=}\text{-INTRO}$$

Figure 2.7: Formation and introduction rules governing = (based on the presentation in The Univalent Foundations Program [91, Appendix A.2]). Γ, *A*, *a* and *b* are arbitrary expressions.

whose elements consist of pairs (x, q) of $x : X$ and $q : Q(x)$. The "dependent" in "dependent sum" refers to the fact that the *type* of the second coordinate q depends on the *value* of the first coordinate x, in contrast to the *Cartesian product* $X \times Y$ of pairs (x, y) with $x : X$ and $y : Y$.

2. We can take the propositional *existential quantifier*

$$(\exists x : X)Q(x)$$

which contains at most one element, representing that there is a valid choice x without specifying such a choice. It is not possible in general to obtain an element of X from a proof of $(\exists x : X)Q(x)$.

The difference between these two choices, too, is central to our work. And again, in our particular case, it turns out that the choice does not matter, as k, if it exists, is unique. Hence we pick the first option, as it is usually easier to work with. The formation and introduction rules for dependent sums are displayed in Figure 2.6.

$$\frac{\begin{array}{c}\Gamma, x : \mathbb{N} \vdash C : \mathcal{U}_i \\ \Gamma \vdash c_0 : C[0/x] \qquad \Gamma, x : \mathbb{N}, y : C \vdash c_s : C[\mathsf{succ}(x)/x] \qquad \Gamma \vdash n : \mathbb{N}\end{array}}{\Gamma \vdash \mathsf{ind}_{\mathbb{N}}(\lambda(x : \mathbb{N}).C, c_0, \lambda(x : \mathbb{N}).\lambda(y : C).c_s, n) : C[n/x]}\ \mathbb{N}\text{-}\textsc{elim}$$

$$\frac{\Gamma, x : \mathbb{N} \vdash C : \mathcal{U}_i \qquad \Gamma \vdash c_0 : C[0/x] \qquad \Gamma, x : \mathbb{N}, y : C \vdash c_s : C[\mathsf{succ}(x)/x]}{\Gamma \vdash \mathsf{ind}_{\mathbb{N}}(\lambda(x : \mathbb{N}).C, c_0, \lambda(x : \mathbb{N}).\lambda(y : C).c_s, 0) \equiv c_0 : C[0/x]}\ \mathbb{N}\text{-}\textsc{comp}_1$$

$$\frac{\begin{array}{c}\Gamma, x : \mathbb{N} \vdash C : \mathcal{U}_i \\ \Gamma \vdash c_0 : C[0/x] \qquad \Gamma, x : \mathbb{N}, y : C \vdash c_s : C[\mathsf{succ}(x)/x] \qquad \Gamma \vdash n : \mathbb{N}\end{array}}{\begin{array}{l}\Gamma \vdash \mathsf{ind}_{\mathbb{N}}(\lambda(x : \mathbb{N}).C, c_0, \lambda(x : \mathbb{N}).\lambda(y : C).c_s, \mathsf{succ}(n)) \\ \qquad\qquad \equiv c_s[n, \mathsf{ind}_{\mathbb{N}}(\lambda(x : \mathbb{N}).C, c_0, \lambda(x : \mathbb{N}).\lambda(y : C).c_s, n)/x, y] \\ \qquad\qquad : C[\mathsf{succ}(n)/x]\end{array}}\ \mathbb{N}\text{-}\textsc{comp}_2$$

Figure 2.8: Induction principle and computation rules governing $\mathbb{N}$ (based on the presentation in The Univalent Foundations Program [91, Appendix A.2]). Γ, C, c_0, c_s and n are arbitrary expressions.

Natural numbers

Now we formalize that n is twice k. We wish to state this as the equality

$$n = 2k.$$

Equalities are formalized via *identity types*, which will define in more detail later. What matters now is that for any type X and any $x, y : X$ we can form the identity type between x and y, denoted $\mathsf{Id}_X(x, y)$, $x =_X y$ or $x = y$. We emphasize that the fact that we can form the type $x =_X y$ does *not* imply that x and y are to be considered equal: after all, the type $x =_X y$ may not contain any elements. For any $x : X$ we have a proof that x is identical to itself, namely the refl constructor. The formation and introduction rules are shown in Figure 2.7.

To formalize $2k$, we define a function $\mathsf{double} : \mathbb{N} \to \mathbb{N}$. At this point we should consider more inference rules associated with $\mathbb{N}$, namely those concerning the *induction principle*. It states that we can define (dependent) functions out of the naturals by specifying what such functions should do for the generators. In our case of defining of defining an *independent* function $\mathsf{double} : \mathbb{N} \to \mathbb{N}$, it suffices to specify a codomain, in our case $\mathbb{N}$, and the value of

1. $\mathrm{double}(0)$, and
2. $\mathrm{double}(\mathrm{succ}(n))$, optionally in terms of $\mathrm{double}(n)$.

These are given by 0 and $\mathrm{succ}(\mathrm{succ}(\mathrm{double}(n)))$, respectively. Hence we may explicitly define double as

$$\mathsf{ind}_{\mathbb{N}}(\lambda(n:\mathbb{N}).\mathbb{N}, 0, \lambda(n:\mathbb{N}).\lambda(m:\mathbb{N}).\,\mathsf{succ}(\mathsf{succ}(m))) : \mathbb{N} \to \mathbb{N}.$$

Remark 2.3.3. A stricter reading of the inference rules says that we have to give a fourth argument to $\mathsf{ind}_{\mathbb{N}}$, namely some $n : \mathbb{N}$, to obtain an element of $\mathbb{N}$ in our above example. Instead we have simply not given a fourth argument to construct an element of type $\mathbb{N} \to \mathbb{N}$. We can justify our reading of the inference rule using a λ-abstraction, or by instead introducing $\mathsf{ind}_{\mathbb{N}}$ as an element of type

$$(\Pi C : \mathbb{N} \to \mathcal{U}_i)(\Pi c_0 : C(0))(\Pi c_s : (\Pi x : \mathbb{N})C(x) \to C(\mathrm{succ}(x)))(\Pi n : \mathbb{N})C(n).$$

Normally induction rules are not introduced as elements of such complex types to avoid depending on Π-types.

Such restrictions of induction principles to independent functions $A \to B$, given by constant type families, are known as *recursion principles*, and we previously discussed the recursion principle for coproducts in Section 2.1.2.

Discussion 2.3.4. There are two further inference rules that essentially say that if you evaluate such inductively defined functions on one of the constructors of $\mathbb{N}$, the expression automatically simplifies in the expected way. In dependent type theories such simplifications are phrased in terms of *judgmental equality* which we denote by $\equiv$. Judgmental equality allows us to transparently identify terms and types that are equal by simplification, such as expanding function evaluations, evaluating inductively defined maps, and substituting a name for its definition. A type theory does not reason about its own judgmental equality, in the sense that we cannot type-theoretically express that two terms are judgmentally equal. If two terms are judgmentally equal, then they can be used interchangeably, and proof assistants such as Coq and Agda recognize this. As such, we will usually invoke judgmental equality implicitly.

Remark 2.3.5. The pattern of four classes of inference rules for a basic type, in this case $\mathbb{N}$, namely

1. a type formation rule that specifies how to obtain the type,
2. introduction rules specifying the generators of the type,
3. an induction rule that allows us to pattern match on elements of the type, and
4. computation rules specifying how the induction rule acts on all the generators,

is typical of so-called *positive types*, which encompass most basic types we discuss except for dependent function types and universes of types. This contrasts with *negative types* such as dependent function types, in which we have the same four classes of inference rules, except that it has a primitive induction rule and a derived introduction rule, rather than vice versa, and *coinductive types*, which have a coinduction rule that tells us how to construct a map *into* it rather than how to eliminate its elements [2].

We give two more examples of positive types. The type $\mathbf{1}$ is specified by the following informal inference rules:

1. the type formation rule that says that $\mathbf{1} : \mathcal{U}_i$ for any universe level i,
2. one introduction rule that says that $\star : \mathbf{1}$,
3. an induction rule that says that in order to define a function $f : (\Pi x : \mathbf{1})C(x)$, it suffices to specify an element of $c : C(\star)$, and
4. a computation rule that says that the function f obtained from the above induction rule satisfies $f(\star) \equiv c$.

The type $\mathbf{0}$ is somewhat degenerate since it has no constructors. Its inference rules consist of:

1. a type formation rule that says that $\mathbf{0} : \mathcal{U}_i$ for any universe level i,
2. no introduction rules since $\mathbf{0}$ is supposed to be an empty type,
3. an induction rule that says that for every type family $C : \mathbf{0} \to \mathcal{U}_i$, we get a function $f : (\Pi x : \mathbf{0})C(x)$ for free, intuitively because there is no possible input to give to f, and
4. no computation rules since there are no corresponding introduction rules.

To conclude, we formalize the theorem as follows. First we define

$$\begin{aligned} \text{double} &:= \mathsf{ind}_{\mathbb{N}}(\lambda(n : \mathbb{N}).\mathbb{N}, 0, \lambda(n : \mathbb{N}).\lambda(m : \mathbb{N}).\, \mathsf{succ}(\mathsf{succ}(m))), \\ \text{isEven}(n) &:= (\Sigma k : \mathbb{N}) n =_{\mathbb{N}} \text{double}(k), \\ \text{isOdd}(n) &:= (\Sigma k : \mathbb{N}) n =_{\mathbb{N}} \mathsf{succ}(\text{double}(k)), \quad \text{and} \\ \text{isEvenOrOdd}(n) &:= \text{isEven}(n) + \text{isOdd}(n), \end{aligned}$$

meaning we define double, isEven, isOdd and isEvenOrOdd to be judgmentally equal to their definition, with := associating more weakly than $=_{\mathbb{N}}$, and then the theorem statement is

$$(\Pi n : \mathbb{N})\, \text{isEvenOrOdd}(n).$$

To prove the theorem means to define an element of this type.

2.3.2 Formalizing the proof

The informal proof works by induction, and the formalized proof will do so as well. Recalling the induction principle for $\mathbb{N}$ in Figure 2.8, we need to provide two elements, respectively with the following types:

$$\begin{aligned} &\text{isEven}(0) + \text{isOdd}(0), \quad \text{and} \\ &(\Pi n : \mathbb{N})\, (\text{isEven}(n) + \text{isOdd}(n)) \to \\ &\quad (\text{isEven}(\mathsf{succ}(n)) + \text{isOdd}(\mathsf{succ}(n)))\,. \end{aligned}$$

Remark 2.3.6. In the second type, the induction case, the outer product $(\Pi n : \mathbb{N})$ binds weakly, and so it should be read as:

$$(\Pi n : \mathbb{N})\big[(\quad \dots \quad) \to (\quad \dots \quad)\big].$$

In words, we need to show that 0 is either even or odd, and that whenever n is either even or odd, then so is $\mathsf{succ}(n)$.

For the base case, we observe that 0 is even. How do we use this fact formally? We need to provide a proof of the base case, and the left disjunct is the one we'd like to show, so we use the inl-constructor for coproducts as in Figure 2.5. It takes an argument, in this case of type $(\Sigma k : \mathbb{N})0 =_{\mathbb{N}} \mathrm{double}(k)$. We construct an element of that type using the introduction rule for dependent sums as in Figure 2.6: we choose 0 for k, in which case $\mathrm{double}(k)$ is judgmentally equal to 0, by using the computation rules of $\mathbb{N}$ as in Figure 2.8, and hence it suffices to show the identity $0 =_{\mathbb{N}} 0$. This identity can be shown using the introduction rule for $=$ in Figure 2.7, which says that $\mathsf{refl}(0) : 0 =_{\mathbb{N}} 0$. So, the full proof of the base case is:

$$\mathsf{inl}((0, \mathsf{refl}(0))).$$

Remark 2.3.7. To verify this proof, that is, to check the judgment

$$\Gamma \vdash \mathsf{inl}((0, \mathsf{refl}(0))) : \mathrm{isEven}(0) + \mathrm{isOdd}(0),$$

we dissect the term according to the inference rules. The outermost constructor is inl, so we try to apply $+$-INTRO$_1$. For this, we need that $\mathrm{isOdd}(0)$ is a well-formed type, which follows from the correctness of our proof statement above, and the correctness of the judgment

$$\Gamma \vdash (0, \mathsf{refl}(0)) : \mathrm{isEven}(0).$$

To check this judgment, we apply Σ-INTRO, and so on. As remarked in Section 2.2, this type checking process is done automatically by proof assistants.

We now show the induction case. Let $n : \mathbb{N}$ be arbitrary, that is, use λ-abstraction to put $n : \mathbb{N}$ in the context. Then we need to define a function that takes an element of a coproduct,

$$\frac{\begin{array}{c}\Gamma, z : (A+B) \vdash C : \mathcal{U}_i \\ \Gamma, x : A \vdash c : C[\mathsf{inl}(x)/z] \qquad \Gamma, y : B \vdash d : C[\mathsf{inr}(y)/z] \\ \Gamma \vdash e : A+B\end{array}}{\Gamma \vdash \mathsf{ind}_{A+B}(\lambda(z : A+B).C, \lambda(x : A).c, \lambda(y : B).d, e) : C[e/z]}\ \textsc{+-elim}$$

$$\frac{\begin{array}{c}\Gamma, z : (A+B) \vdash C : \mathcal{U}_i \qquad \Gamma, x : A \vdash c : C[\mathsf{inl}(x)/z] \qquad \Gamma, y : B \vdash d : C[\mathsf{inr}(y)/z] \\ \Gamma \vdash a : A\end{array}}{\begin{array}{l}\Gamma \vdash \mathsf{ind}_{A+B}(\lambda(z : A+B).C, \lambda(x : A).c, \lambda(y : B).d, \mathsf{inl}(a)) \equiv c[a/x] \\ \qquad : C[\mathsf{inl}(a)/z]\end{array}}\ \textsc{+-comp}_1$$

$$\frac{\begin{array}{c}\Gamma, z : (A+B) \vdash C : \mathcal{U}_i \qquad \Gamma, x : A \vdash c : C[\mathsf{inl}(x)/z] \qquad \Gamma, y : B \vdash d : C[\mathsf{inr}(y)/z] \\ \Gamma \vdash b : B\end{array}}{\begin{array}{l}\Gamma \vdash \mathsf{ind}_{A+B}(\lambda(z : A+B).C, \lambda(x : A).c, \lambda(y : B).d, \mathsf{inr}(b)) \equiv d[b/y] \\ \qquad : C[\mathsf{inr}(b)/z]\end{array}}\ \textsc{+-comp}_2$$

Figure 2.9: Induction principle and computation rules governing + (based on the presentation in The Univalent Foundations Program [91, Appendix A.2]). Γ, A, B, C, a, b, c, d and e are arbitrary expressions.

and outputs an element of another coproduct. We use the induction principle for coproducts, displayed in Figure 2.9. It says that we need to provide two maps:

$$\text{isEven}(n) \to \big(\text{isEven}(\text{succ}(n)) + \text{isOdd}(\text{succ}(n))\big)$$

and

$$\text{isOdd}(n) \to \big(\text{isEven}(\text{succ}(n)) + \text{isOdd}(\text{succ}(n))\big).$$

For the first map, if n is even, then surely $\text{succ}(n)$ is odd. But how to show this? Using the induction principle of Σ-types, loosely speaking, we may assume that the input of type $\text{isEven}(n)$ is of the form (k, p) with $k : \mathbb{N}$ and $p : n =_{\mathbb{N}} 2k$. We wish to find $k' : \mathbb{N}$ and $p' : \text{succ}(n) =_{\mathbb{N}} \text{succ}(\text{double}(k))$. Setting k' to be k should do the trick, after all, if $x =_{\mathbb{N}} y$ then surely $\text{succ}(x) =_{\mathbb{N}} \text{succ}(y)$. But how to obtain p'?

The more general principle that we need is that if we have $f : X \to Y$ and $p : x =_X x'$, with in our case succ as f, then we should be able to obtain an element of $f(x) =_Y f(x')$. To obtain this, we discuss the induction principle of = in more detail.

$$\frac{\begin{array}{c}\Gamma, x : A, y : A, p : x =_A y \vdash C : \mathcal{U}_i \\ \Gamma, z : A \vdash c : C[z, z, \mathsf{refl}(z)/x, y, p] \qquad \Gamma \vdash a : A \qquad \Gamma \vdash b : A \qquad \Gamma \vdash p' : a =_A b\end{array}}{\Gamma \vdash \mathsf{ind}_{=_A}(\lambda(x : A).\lambda(y : A).p.C, \lambda(z : A).c, a, b, p') : C[a, b, p'/x, y, p]} \text{=-ELIM}$$

$$\frac{\Gamma, x : A, y : A, p : x =_A y \vdash C : \mathcal{U}_i \qquad \Gamma, z : A \vdash c : C[z, z, \mathsf{refl}(z)/x, y, p] \qquad \Gamma \vdash a : A}{\begin{array}{c}\Gamma \vdash \mathsf{ind}_{=_A}(\lambda(x : A).\lambda(y : A).\lambda(p : x =_A y).C, \lambda(z : A).c, a, a, \mathsf{refl}(a)) \equiv c[a/z] \\ : C[a, a, \mathsf{refl}(a)/x, y, p]\end{array}} \text{=-COMP}$$

Figure 2.10: Induction principle and computation rules governing = (based on the presentation in The Univalent Foundations Program [91, Appendix A.2]). Γ, A, C, a, b, p' and c are arbitrary expressions.

The induction and computation rules governing = are shown in Figure 2.10. The induction principle for identity types says that to construct a map of type

$$(\Pi x, y : A)(\Pi p : x =_A y)C(x, y, p)$$

for some type family

$$C : (\Pi x, y : A)x =_A y \to \mathcal{U}_i,$$

it suffices to, for every $x : A$, provide a value of $C(x, x, \mathsf{refl}(x))$.

We emphasize that the induction principle does *not* tell us how to construct an element of

$$(\Pi p : x =_A y)D(p)$$

with

$$D : x =_A y \to \mathcal{U}_i.$$

The latter could be argued to follow the same pattern as the induction principle for $\mathbb{N}$ and +: after all, the only way to construct an element of identity types is $\mathsf{refl}(x) : x =_A x$. However, crucially, it is not the individual types $x =_A y$ that are defined by a generator, but the *type family* of identity types between any two elements $x, y : A$ in a fixed type A, which is defined by the reflexivity generators. Indeed, even though the only generator specified by the identity types is refl, we cannot prove that it is the only element of the type $x =_A x$. In fact, in the presence of univalence, we can define types whose identity types have elements that are themselves not identical to reflexivity.

Hence we see that we must generate an element of $f(x) =_Y f(x')$ given an identity $x =_Y x'$ for *arbitrary* $x, x' : X$. So what we want is an element of

$$(\Pi x, x' : X)(\Pi p : x =_X x')f(x) =_Y f(x').$$

This fits the induction principle above by setting

$$C(x, x', p) :\equiv (f(x) =_Y f(x')).$$

Hence what we need is, for any $x : X$, an element of $C(x, x, \mathsf{refl}(x)) \equiv (f(x) =_Y f(x))$, which we can give as $\mathsf{refl}(f(x))$. Concretely, the element is

$$\begin{aligned}&\mathsf{ind}_{=_X}(\lambda(x : X).\lambda(x' : X).\lambda(p : x =_X x').(f(x) =_Y f(x')), \lambda(x : X).\, \mathsf{refl}(f(x)))\\ &\qquad : (\Pi x, x' : X)(\Pi p : x =_X x')f(x) =_Y f(x').\end{aligned}$$

As we use this principle more often, we write $\mathrm{ap}_{f,x,x'}$ for this element, whose type is

$$x =_X x' \to f(x) =_Y f(x').$$

By the above argument, we know that $x =_{\mathbb{N}} y$ implies $\mathrm{succ}(x) =_{\mathbb{N}} \mathrm{succ}(y)$. This completes the proof of

$$\mathrm{isEven}(n) \to \big(\mathrm{isEven}(\mathrm{succ}(n)) + \mathrm{isOdd}(\mathrm{succ}(n))\big).$$

After all: by the induction principle of Σ-types, we may assume the input to be of the form (k, p) with $k : \mathbb{N}$ and $p : n =_{\mathbb{N}} \mathrm{double}(k)$. Thus we can output $\mathsf{inr}(k, \mathrm{ap}_{\mathrm{succ},n,\mathrm{double}(k)}(p))$. So we have the element e of the above type defined by the equation

$$e((k, p)) :\equiv \mathsf{inr}(k, \mathrm{ap}_{\mathrm{succ},n,\mathrm{double}(k)}(p)).$$

Next, assume that we have a proof of $\mathrm{isOdd}(n)$, then we wish to show

$$\big(\mathrm{isEven}(\mathrm{succ}(n)) + \mathrm{isOdd}(\mathrm{succ}(n))\big).$$

This should be true because if n is odd, then $\mathrm{succ}(n)$ is even. Again, we may assume the witness of $\mathrm{isOdd}(n)$ to be of the form (k, p) with $k : \mathbb{N}$ and $p : n =_{\mathbb{N}} \mathrm{succ}(\mathrm{double}(k))$. Hence we set $k' :\equiv \mathrm{succ}(k)$ and we wish to show that $\mathrm{succ}(n) =_{\mathbb{N}} \mathrm{double}(k')$, namely, that n is even. Now $\mathrm{double}(k') = \mathrm{succ}\,\mathrm{succ}(\mathrm{double}(k))$ by the computation rule for double, which was defined using the induction principle for $\mathbb{N}$. Hence showing that $\mathrm{succ}(n) =_{\mathbb{N}} \mathrm{double}(k')$ is equivalent to showing $\mathrm{succ}(n) =_{\mathbb{N}} \mathrm{succ}(\mathrm{succ}(\mathrm{double}(k)))$, which is true by $\mathrm{ap}_{\mathrm{succ},n,\mathrm{succ}(\mathrm{double}(k))}(p)$. In conclusion, we have the element

$$o : \mathrm{isOdd}(n) \to \big(\mathrm{isEven}(\mathrm{succ}(n)) + \mathrm{isOdd}(\mathrm{succ}(n))\big)$$

defined by the equation

$$o((k, p)) :\equiv \mathsf{inl}((\mathrm{succ}(k), \mathrm{ap}_{\mathrm{succ},n,\mathrm{succ}(\mathrm{double}(k))}(p))).$$

This completes the induction step of our proof that every natural number is either even or odd, and hence we have shown the theorem.

2.4 Case study: surjective maps, images of maps

We now discuss the following theorem, whose statement in set theories is clear, but whose type-theoretic translation needs discussion.

Theorem. *Let $f : X \to Y$ be a map. The following are equivalent:*

1. *For every element of Y there exists a pre-image in X.*

2. *The image of f is equal to Y.*

We will postpone the proof until after the formalization of the theorem, as the proof is intended to be straightforward. The theorem statement, however, requires some attention.

2.4.1 Formalizing the theorem statement

First of all, the statement concerns a map $f : X \to Y$ between arbitrary types X, Y. So we start with three Π-types that express this universal quantification:

$$(\Pi X, Y : \mathcal{U})(\Pi f : X \to Y) \ldots$$

The claim about some particular f, namely that two conditions are equivalent, is a biconditional, which is formalized as the product of two implications:

$$(\Pi X, Y : \mathcal{U})(\Pi f : X \to Y)(\ldots \to \ldots) \times (\ldots \to \ldots),$$

where $A \times B$ means $(\Sigma x : A)B$, formalized using a Σ-type with a constant type family $B' : A \to \mathcal{U}$ defined by $B'(a) :\equiv B$. Such "independent sum types" are more typically referred to as *Cartesian products* or *product types.*

Existence of pre-images

A pre-image of $y : Y$ under $f : X \to Y$ is an element of $(\Sigma x : X) f(x) = y$, also known as the *fiber* of f over y.

How do we formalize that $y : Y$ has a pre-image under f in X? We recall the two ways to formalize existential quantification in univalent type theory:

1. We can form the *dependent sum*

   $$(\Sigma x : X) Q(x)$$

   whose elements consist of pairs (x, q) of $x : X$ and $q : Q(x)$. The "dependent" in "dependent sum" refers to the fact that the *type* of the second coordinate q depends on the *value* of the first coordinate x, in contrast to the *Cartesian product* $X \times Y$ of pairs (x, y) with $x : X$ and $y : Y$.

2. We can take the propositional *existential quantifier*

   $$(\exists x : X) Q(x)$$

which contains at most one element, representing that there is a valid choice x without specifying such a choice. It is not possible in general to obtain an element of X from a proof of $(\exists x : X)Q(x)$.

The elements of the formalization following the first choice, that is, the elements of

$$(\Pi y : Y)(\Sigma x : X)f(x) = y$$

correspond to, for each $y : Y$, a *choice* of pre-image in X. In particular, from an element of $(\Pi y : Y)(\Sigma x : X)f(x) = y$ we can obtain a function $g : Y \to X$ which is right inverse to f. Hence, we have not formalized when f is surjective, but when f is a *retraction*.

We would like to fix this by hiding the choice of pre-images, so that we do not obtain a right inverse to f. The second choice for the existential quantification does this, so we describe it in some more detail below.

In any case, the correct formalization of the first statement is:

$$(\Pi y : Y)(\exists x : X)f(x) = y.$$

Propositions

Definition 2.4.1. A *proposition* is a type P all whose elements are identical, which is expressed type-theoretically as

$$\mathrm{isHProp}(P) :\equiv (\Pi p, q : P)(p =_P q).$$

We have the type

$$\mathrm{HProp} :\equiv (\Sigma P : \mathcal{U})\,\mathrm{isHProp}(P)$$

of all propositions, and we identify elements of HProp with their underlying type, that is, their first projection. The letter *'H'* stands for *homotopy*, which we briefly touch on in Section 2.5.4.

Although we generally speaking abstain from specifying which universe $\mathcal{U}$ we are referring to, in this case we emphasize that, in reality, the above definition assigns a type HProp to each choice of universe $\mathcal{U}$. These respective types of propositions are, in the first instance, not provable to be equal, and we return to this point in Section 2.6.1.

$$\frac{\Gamma \vdash A : \mathcal{U}_i}{\Gamma \vdash \|A\| : \mathcal{U}_i}\ \|\cdot\|\text{-}\textsc{form} \qquad\qquad \frac{\Gamma \vdash a : A}{\Gamma \vdash |a| : \|A\|}\ \|\cdot\|\text{-}\textsc{intro}_1$$

$$\frac{\Gamma \vdash e : \|A\| \qquad \Gamma \vdash e' : \|A\|}{\Gamma \vdash \mathsf{sq}(e, e') : e =_{\|A\|} e'}\ \|\cdot\|\text{-}\textsc{intro}_2$$

$$\frac{\begin{array}{c}\Gamma, z : \|A\| \vdash C : \mathcal{U}_i \qquad \Gamma, z : \|A\| \vdash p : \mathrm{isHProp}(C) \\ \Gamma, x : A \vdash c : C[|x|/z] \qquad \Gamma \vdash e : \|A\|\end{array}}{\Gamma \vdash \mathsf{ind}_{\|A\|}(\lambda(z : \|A\|).C, \lambda(z : \|A\|).p, \lambda(x : A).c, e) : C[e/z]}\ \|\cdot\|\text{-}\textsc{elim}$$

$$\frac{\begin{array}{c}\Gamma, z : \|A\| \vdash C : \mathcal{U}_i \qquad \Gamma, z : \|A\| \vdash p : \mathrm{isHProp}(C) \\ \Gamma, x : A \vdash c : C[|x|/z] \qquad \Gamma \vdash a : A\end{array}}{\Gamma \vdash \mathsf{ind}_{\|A\|}(\lambda(z : \|A\|).C, \lambda(z : \|A\|).p, \lambda(x : A).c, |a|) \equiv c[a/x] : C[|a|/z]}\ \|\cdot\|\text{-}\textsc{comp}$$

Figure 2.11: The inference rules governing $\|\cdot\|$. A, C, a, c, e, e' and p are arbitrary expressions.

Propositions are used to model truth values in type theory, to contrast with arbitrary types, which can have arbitrarily many elements. We will use the word "proof" also to refer to type-theoretic constructions, including when such constructions are not unique. Additionally, we use the word "proposition" to mean a mathematical statement of neutral importance.

In univalent type theory, it is assumed that every type X has a propositional truncation $\|X\|$.

Axiom 2.4.2. For every type X, there is a propositional truncation $\|X\|$ that satisfies the inference rules in Figure 2.11.

Definition 2.4.3. The *propositional truncation* of a type X is a proposition $\|X\|$ together with a *truncation map* $|\cdot| : X \to \|X\|$ such that for any other proposition Q, given a map $g : X \to Q$, we obtain a map $h : \|X\| \to Q$.

> *Remark* 2.4.4. The uniqueness of the obtained map $\|X\| \to Q$ follows from the fact that Q is a proposition, and function extensionality, which is discussed in Section 2.5.1.

The propositional truncation $\|X\|$ of a type X is a proposition. We may say, quite simply, that we have a constructor sq which is a proof that the type $\|X\|$ is a squashed to be a proposition: it takes two elements of $\|X\|$ and gives a proof that they are identical, i.e. squashed together.

The recursion principle of Definition 2.4.3 may be expressed diagrammatically as the unique existence of a vertical map in the following diagram.

$$\begin{tikzcd}
X \arrow[r, "|\cdot|"] \arrow[rd] & \|X\| \arrow[d, dashed, "\exists!"] \\
& Q
\end{tikzcd}$$

Corresponding to this recursion principle, we may phrase an induction principle as follows. Given a type family $B : \|X\| \to \mathcal{U}$, such that $B(t)$ is a proposition for each $t : \|X\|$, and given the data $f_{|\cdot|} : (\Pi x : X)B(|x|)$, we obtain a map $f : (\Pi t : \|X\|)B(t)$ satisfying the computation rule $f(|x|) \equiv f_{|\cdot|}(x)$ for each $x : X$.

Propositional truncations can be defined as higher-inductive types, namely, as a type that is freely generated by both constructors and constructors of its (iterated) identity types, which are also discussed in Section 2.5.4. Propositional truncations can alternatively be constructed via impredicative encodings assuming propositional resizing [8].

One motivation of propositional truncation is that in toposes, the truncation $\|X\|$ of an object X can be computed as the image of the unique map $X \to \mathbf{1}$ to the terminal object. Note that toposes have Σ and Π as left and right adjoint to the pullback functor, and $(\exists x : A)\phi(x)$ is isomorphic to $\|(\Sigma x : A)\phi(x)\|$ in any topos, and in fact we can take this existential quantifier as primitive rather than truncation. In setoid-based mathematics, the truncation of a setoid is given by the same underlying set of elements, but with the chaotic equivalence relation that relates all elements.

Univalent logic, the logic of propositions, is then the following generalization of the logic of truth values in toposes, namely elements of the subobject classifier Ω.

Definition 2.4.5. *Univalent logic* is defined by the following, where P, Q : HProp and $R : X \to$ HProp [91, Definition 3.7.1]:

$$\top :\equiv \mathbf{1}$$

$$\bot :\equiv \mathbf{0}$$

$$\begin{aligned}
P \wedge Q &:= P \times Q \\
P \Rightarrow Q &:= P \to Q \\
P \Leftrightarrow Q &:= P = Q \\
\neg P &:= P \to \mathbf{0} \\
P \vee Q &:= \|P + Q\| \\
(\forall x : X) R(x) &:= (\Pi x : X) R(x) \\
(\exists x : X) R(x) &:= \|(\Sigma x : X) R(x)\|
\end{aligned}$$

We use the above notation when the types involved are indeed propositions. For example, we write $X \Rightarrow Y$ when X and Y are propositions, and $X \to Y$ if this has not been established.

Propositions are types that have at most one element. The truncation of a type X is a type which has exactly one point if X has any number of elements: we think of $\|X\|$ as $\{\, \star \mid \exists x \in X \,\}$. However, with the way we have set things up, this is not a very informative definition as $\exists$ is in turn defined in terms of $\| \cdot \|$.

Discussion 2.4.6. We revisit the propositional logic example of Section 2.1. We may formalize the statement of Section 2.1 in univalent logic as

$$(P \vee Q) \Rightarrow (\neg P \Rightarrow Q).$$

To prove this statement, as we will see in Lemma 2.5.21, $\neg P \to Q$ is a proposition, and so we may use the elimination rule for propositional truncations. It says that to prove the above, it suffices to construct an element of

$$(P + Q) \to (\neg P \to Q),$$

which in turn can be constructed using the recursion principle for coproducts, for which it suffices to show $P \to (\neg P \to Q)$ and $Q \to (\neg P \to Q)$, which can be done as in Section 2.1.

Hence, this formulation of propositional logic in propositions in the sense of Definition 2.4.1 also holds.

Definition 2.4.7. We refer to types that are propositions as *properties*. We refer to types that potentially have several witnesses as *structure*. The claim "there exists an A satisfying B" is to be interpreted by $\exists$, and the claims "we can find A satisfying B", "we have A satisfying B" and "we have A equipped with a B" are to be interpreted by Σ.

Example 2.4.8. Suppose we want to state that we have a method for computing roots of polynomials over a certain ring R. This statement should be formalized as

$$(\Pi f : R[X])(\Sigma x : R) f x = 0$$

rather than

$$(\forall f : R[X])(\exists x : R) f x = 0,$$

as the latter exposes the existence as a truth value rather than as a witness.

Even though the elimination rule in Definition 2.4.1 only constructs maps into propositions, we can sometimes obtain witnesses from existence.

Definition 2.4.9. A *decidable* proposition is a proposition P such that $P + \neg P$. We have the collection

$$\mathrm{DHProp} := (\Sigma P : \mathrm{HProp}) P + \neg P$$

of decidable propositions. We identify elements of DHProp with their underlying proposition, and hence with their underlying types.

> *Remark* 2.4.10. If P and Q are decidable, then so is $P \wedge Q$, and we use this fact in later developments.

Example 2.4.11 (Escardó [40], [44], [91, Exercise 3.19])**.** Let $P : \mathbb{N} \to \mathrm{DHProp}$. If $(\exists n : \mathbb{N}) P(n)$ then we can construct an element of $(\Sigma n : \mathbb{N}) P(n)$. This element is constructed as the least n satisfying P in Theorem 6.5.1.

Images of maps

What is the image of a map $f : X \to Y$? It is the collection of elements of Y which are mapped to by f from some element of X. As in the definition of surjectivity, we have to make sure not to include a *choice* of pre-image. Indeed, the type

$$(\Sigma y : Y)(\Sigma x : X) f(x) = y$$

is isomorphic to X, in the sense that there are maps back and forth, with both compositions being pointwise equal to the identity map, and so does not represent the image of f. We will make this more precise in Example 2.5.11. Again, to correct this, we hide the choice of pre-image using the existential quantifier:

$$\mathrm{im}(f) :\equiv (\Sigma y : Y)(\exists x : X) f(x) = y$$

Perhaps surprisingly from a set-theoretic point of view, for our purposes it would be wrong to formalize the claim that the image of f is equal to Y as the identity type

$$\mathrm{im}(f) =_{\mathcal{U}} Y.$$

Indeed, there are non-surjective maps for which this is true, such as our map double : $\mathbb{N} \to \mathbb{N}$ above: the image of double is the collection of even natural numbers, and there is an isomorphism between the type of even naturals and the type of all naturals. Univalence, discussed in Section 2.5.4, tells us in particular that from an isomorphism between types we can obtain an identification between types. Hence, the image of double, as a type, is identical to its codomain $\mathbb{N}$, even though double is not surjective.

This behavior is a feature, rather than a bug, of type theory: elements of types are seen intensionally, i.e. they have no internal structure that can be observed by other types. In contrast, in material set theories, the elements of sets are themselves sets, and in such systems it makes sense at least on a formal level to ask, for example, whether $\{5, \{\exp\}\} \in \frac{7}{3}$.

In our example of the double function, we would like to fix the formalization by somehow specifying that we would like to see the collection of even natural numbers as a subset or subtype of the natural numbers: after all, as a subset of $\mathbb{N}$, the set of even naturals is not equal to the set of all naturals.

After we have formalized subsets below, we can formalize the second claim in the theorem statement as $\mathrm{im}(f) \subseteq Y \wedge Y \subseteq \mathrm{im}(f)$, where $\mathrm{im}(f)$ and Y are seen as subsets of Y.

Subtypes

What is a subtype of a fixed type $X : \mathcal{U}$?

1. From a logical perspective, one could define a subtype of X to be a predicate on X which gives a truth value for every element of X.

2. From a categorical perspective, one could define a subtype to be a type Y, together with an inclusion map from Y to X.

Both notions can be made precise in univalent type theory, and we will show in Lemma 2.6.3 that they are equivalent in an appropriate sense. For now, we focus on the former perspective.

A *predicate* on a type X is a function $A : X \longrightarrow \mathrm{HProp}$. Given such a predicate, we can see A as a subset of X by defining

$$x \in A :\equiv A(x).$$

We can define $A \subseteq B$ for $A, B : X \longrightarrow \mathrm{HProp}$ by

$$(\forall x : X) x \in A \Rightarrow x \in B.$$

The powerset of X may be defined as, simply, $X \longrightarrow \mathrm{HProp}$: its elements correspond to subsets of X.

Remark 2.4.12. Recalling that HProp is defined in terms of a universe of types $\mathcal{U}$, and recalling that there is not one universe $\mathcal{U}$, but, for example, a hierarchy of universes $\mathcal{U}_0 : \mathcal{U}_1 : \mathcal{U}_2 : \ldots$, for each universe level i we have a different notion of subset of X.

Conclusion

Recalling the definition of the image of a map above,

$$\mathrm{im}(f) :\equiv (\Sigma y : Y)(\exists x : X) f(x) = y,$$

we observe that the inclusion map to X should indeed be the projection map pr_0, and that the corresponding predicate $A : X \longrightarrow \mathrm{HProp}$ is $A(x) = (\exists x : X) f(x) = y$. Explicitly, the second part of the theorem statement says that the subtype Y of Y is wholly contained in the subtype $\mathrm{im}(f)$ of Y, that is, $Y \subseteq \mathrm{im}(f)$. The subtype Y of Y is represented by the predicate which is constantly true. Hence we obtain the formalization as:

$$(\Pi y : Y)\top \to (\exists x : X)f(x) = y.$$

Then the full theorem statement is:

$$((\Pi y : Y)(\exists x : X)f(x) = y) \Leftrightarrow ((\Pi y : Y)\top \Rightarrow (\exists x : X)f(x) = y)\,.$$

2.4.2 Formalizing the proof

We have to prove two implications. First, consider

$$((\Pi y : Y)(\exists x : X)f(x) = y) \to ((\Pi y : Y)\top \Rightarrow (\exists x : X)f(x) = y)\,.$$

We prove this by several λ-abstractions, so let $g : (\Pi y : Y)(\exists x : X)f(x) = y$, let $y : Y$ and $t : \top$. Then $g(y) : (\exists x : X)f(x) = y$, as required.

For the other implication, we need to show

$$((\Pi y : Y)\top \Rightarrow (\exists x : X)f(x) = y) \to ((\Pi y : Y)(\exists x : X)f(x) = y)\,.$$

Let $h : (\Pi y : Y)\top \Rightarrow (\exists x : X)f(x) = y$, let $y : Y$. Recalling the generator $\star : \mathbf{1}$, we have $h(y, \star) : (\exists x : X)f(x) = y$, as required.

The full proof is thus:

$$(\lambda g.\lambda y.\lambda t.g(y), \lambda h.\lambda y.h(y, \star)).$$

2.5 Extensionality

Univalent type theory aims to give the identity types a central role. In contrast, in traditional set theories, where sets come equipped with a notion of equality specified by first-order logic, many objects we like to think of as being equal, such as isomorphic groups, are in fact distinct sets. In this section, we give a taste of the central role of the identity type in univalent type theory.

2.5.1 Function extensionality

We would like to say that two functions are identical as elements of the function type if and only if they are pointwise equal. However, it is not clear at first sight that the two claims of equality-as-functions and pointwise equality we are comparing are propositions, and in fact in general they are not, as we discuss below. This hints at the fact that the exact phrasing of the desired equivalence between equality of functions and pointwise equality, namely *function extensionality*, requires some choices. Some variations can be found in Garner [47] and Lumsdaine [69].

Nonetheless, an implication from equality of functions to pointwise equality of functions can be shown with the techniques we already discussed so far.

Lemma 2.5.1. *Let $X, Y : \mathcal{U}$ be arbitrary types. For $f, g : X \to Y$, given an element of $f = g$ we can prove $(\Pi x : X) f(x) = g(x)$.*

Proof. By λ-abstraction we may define an evaluation map

$$\mathrm{ev}' :\equiv \lambda(x : X).\lambda(h : X \to Y).h(x) : X \to (X \to Y) \to Y.$$

Now for any $x : X$, we have $\mathrm{ev}'(x) : (X \to Y) \to Y$ and hence

$$\mathrm{ap}_{\mathrm{ev}'(x), f, g} : f = g \to f(x) = g(x),$$

so that we have the full proof

$$\lambda f.\lambda g.\lambda p.\lambda x.\mathrm{ap}_{\mathrm{ev}'(x),f,g}(p) : (\Pi f, g : X \to Y) f = g \to (\Pi x : X) f(x) = g(x). \qquad \square$$

Similarly, for dependent functions $f, g : (\Pi x : X) Y(x)$ with $f = g$ we get a pointwise identity $(\Pi x : X) f(x) = g(x)$:

Lemma 2.5.2. *Let $X, Y : \mathcal{U}$ be arbitrary types. For $f, g : (\Pi x : X) Y(x)$, given an element of $f = g$ we can prove $(\Pi x : X) f(x) = g(x)$.*

This hints at a dependent version of the ap principle, which can be stated in terms of *dependent identifications* as in Definition 4.5.17.

In both the dependent and independent case, we would like to conversely say that from $(\Pi x : X) f(x) = g(x)$ we may conclude $f = g$, which is not provable in dependent type theory, because there are also type theories in which equality of functions is thought of as two functions being implemented by the same algorithm [76]. In that sense, extensionality can be seen as a feature we can choose to have, or not, when setting up the type theory.

Definition 2.5.3. *Naive independent function extensionality* is the claim that Lemma 2.5.1 has a converse: for all $X, Y ; \mathcal{U}$ and $f, g : X \to Y$, if $(\Pi x : X) f(x) = g(x)$ then $f = g$.

Perhaps surprisingly, it may happen that this requirement is not a proposition: there may be different ways to exhibit this inverse implication [91, Theorem 4.1.3], just like there may be different ways to prove the above lemmas. This means that we have not unambiguously specified our wish. In particular this happens when X and Y are not *sets*.

Definition 2.5.4. A *set* is a type X whose identity types are propositions, that is:

$$\mathrm{isSet}(X) :\equiv (\Pi x, y : X) \, \mathrm{isHProp}(x =_X y).$$

In our work on constructive analysis, we will mainly be concerned with sets. But the presence of types which are not sets is something we will have to deal with in our type-theoretic foundations.

We can disambiguate our search for a converse of Lemma 2.5.1 by defining a *proposition* that expresses that the specific map constructed in that lemma is invertible in the appropriate sense. This appropriate notion of having both left- and right-sided inverses, namely that it is an *equivalence*, is discussed in Section 2.5.4.

However, since we are mainly concerned with sets, any converse of Lemma 2.5.1 for arbitrary types will do, since on those types that are sets any converse implication is unique. Our focus on naive function extensionality is additionally motivated by the surprising but non-trivial fact that an *arbitrary converse* of Lemma 2.5.1, even for types that are not sets, can be used to show that the particular implication from equality of functions to pointwise equality constructed in Lemma 2.5.1, or even its variant for dependent function in Lemma 2.5.2, is an *equivalence* [69].

Sections 2.5.3 and 2.5.5 discuss two ways to construct a converse to Lemma 2.5.1.

2.5.2 Contractibility

As a preliminary to the next subsection, we discuss types that have exactly one point, namely *contractible* types.

Definition 2.5.5. A type $X : \mathcal{U}$ is *contractible* if it is a pointed proposition, that is:

$$\mathrm{isContr}(X) :\equiv X \times \mathrm{isHProp}(X).$$

The first projection of an element of $\mathrm{isContr}(X)$, that is, the corresponding element of X, is referred to as the *center of contraction.*

We use contractibility to formalize claims of "existence and uniqueness", for example in Theorem 5.2.1, Theorem 5.3.4, and an equivalent definition of the notion of equivalences in Section 2.5.4.

Despite Definition 2.5.5 suggesting to be data-like in the sense that the chosen point of X seems part of the data, contractibility is a proposition. To show this, we first show that being a proposition itself is a proposition.

Lemma 2.5.6. *For an arbitrary type $X : \mathcal{U}$, we have* $\mathrm{isHProp}(\mathrm{isHProp}(X))$.

Proof. Let $p, q : \mathrm{isHProp}(X)$, that is, $p, q : (\Pi x, y : X)x = y$. We wish to show $p = q$. As is typical, to show the equality of the dependent functions p and q, we use function extensionality. One application of function extensionality says that in order to show that p and q are identical, it suffices to show, for every $x : X$, that $p(x) = q(x)$. This, in turn, can be shown by another application of function extensionality, so that it now suffices to show, for every $x, y : X$, that $p(x, y) = q(x, y)$. Note, however, that this is an identity in X itself, which we know to be a proposition by p (or q). Hence we obtain the required identification $p(x, y) = q(x, y)$. □

Corollary 2.5.7. *For an arbitrary type $X : \mathcal{U}$, we have* $\mathrm{isHProp}(\mathrm{isContr}(X))$.

Proof. Let $w, z : \text{isContr}(X)$, that is, $w, z : X \times \text{isHProp}(X)$. Using the induction principles of product types and identity types, in order to show an equality in a product type, it suffices to show equality coordinate-wise. The second coordinates of w and z are identical because of the above lemma, and the first coordinates are identical because the second coordinate of either w or z says that X is a proposition. □

We now give two examples of contractible types.

Lemma 2.5.8. *The unit type* $\mathbf{1}$ *is contractible.*

Proof. To show that $\mathbf{1}$ is a proposition, let $x, y : \mathbf{1}$. By the induction principle of $\mathbf{1}$, we may assume both x and y to be judgmentally equal to $\star$, so that it suffices to show the identity $\star = \star$, which we get by reflexivity.

We have the center of contraction $\star : \mathbf{1}$. □

The second example is that of *singleton types.* In order to show that such singleton types are contractible, we first develop a one-sided variant of the induction principle for identity types.

Lemma 2.5.9. *Let X be any type, and $x : X$ a point. Given a type family $C : (\Pi y : X) x = y \to \mathcal{U}$ and an element $c : C(x, \mathsf{refl}(x))$, we get an element $f : (\Pi y : X)(\Pi p : x = y) C(y, p)$ satisfying the computation rule $f(x, \mathsf{refl}(x)) \equiv c$ judgmentally.*

Proof. We apply induction principle of = as in Figure 2.10 to

$$D(z, w, q) :\equiv (\Pi C' : (\Pi y : X) z = y \to \mathcal{U}) C'(z, \mathsf{refl}(z)) \to C'(w, q),$$
$$d(z) :\equiv \lambda C'.\lambda c'.c',$$

so that we obtain $f' : (\Pi z, w : X)(\Pi q : z = w) D(z, w, q)$ with $f'(z, z, \mathsf{refl}(z)) \equiv d(z)$. Then we set $f(y, p) :\equiv f'(x, y, p, C, c)$, so that $f(x, \mathsf{refl}(x)) \equiv f'(x, x, \mathsf{refl}(x), C, c) \equiv d(z, C, c) \equiv c$. □

Lemma 2.5.10. *Let $X : \mathcal{U}$ and $x : X$. The* singleton type *$(\Sigma y : X) x = y$ is contractible.*

Proof. We have the center of contraction $(x, \mathsf{refl}(x))$. To show that $(\Sigma y : X) x = y$ is a proposition, let (y, p) and (y', p') be elements of $(\Sigma y : X) x = y$. To show that (y, p) and (y', p') are identical, apply based path induction twice on the statement

$$(\Pi y : X)(\Pi p : x = y)(\Pi y' : X)(\Pi p' : x = y')(y, p) = (y', p'),$$

so that it suffices to show that $(x, \mathsf{refl}(x)) = (x, \mathsf{refl}(x))$, which holds by reflexivity. □

Example 2.5.11. As an application of the contractibility of singletons, we consider the naive formalization $(\Sigma y : Y)(\Sigma x : X)f(x) = y$ of the image of a function $f : X \to Y$, discussed in Section 2.4.1. In a sense we will discuss in more detail in Section 2.5.4, this type is *equivalent* to the type $(\Sigma x : X)(\Sigma y : Y)f(x) = y$, where we have simply swapped the order of the dependent sums. Now the inner dependent sum $(\Sigma y : Y)f(x) = y$ is a singleton type, and hence, as can be made precise later, equivalent to $\mathbf{1}$, so that the type is equivalent to $(\Sigma x : X)\mathbf{1}$, that is, to X.

2.5.3 Computation rules for higher-inductive types

As mentioned in Section 2.4.1, we can define the propositional truncation as a *higher-inductive type*: a type that is freely generated by both constructors of the type, and constructors of (iterated) identity types of that type. A propositional truncations $\|X\|$ is generated by the following two constructors:

1. $|\cdot| : X \to \|X\|$, and
2. $\mathsf{sq} : (\Pi s, t : \|X\|)s =_{\|X\|} t$.

This raises the question of which such lists of constructors are admissible, that is, can be interpreted semantically. This has been answered to some extent in Lumsdaine and Shulman [70].

We recall from Remark 2.3.5 that the constructors of an inductively defined type should come with corresponding computation rules. For higher-inductive types, it is not entirely clear how this should be done, since computation rules corresponding to the identifications, sq in this case, are naively stated in terms of the *defined* element ap [91, Notes to Chapter 6], rather than in terms of more primitive notions, which are better understood. Nonetheless, for the propositional truncation, following The Univalent Foundations Program [91], we have given a computation rule for $|\cdot|$ as a *judgmental* equality in Figure 2.11. This has the perhaps unintuitive consequence that we can write a mysterious term myst with $\mathrm{id}_{\mathbb{N}} \equiv \mathrm{myst} \circ |\cdot|$ that seems to undo the information hiding of propositional truncation [61].

With this computation rule for $|\cdot|$, we can prove naive function extensionality, following Lumsdaine [69] and Escardó [41]. As a preliminary to this, note that the type $I :\equiv \|2\|$, namely the truncation of the Booleans, which has two constructors $\mathtt{tt}, \mathtt{ff} : \mathbf{2}$, satisfies the following recursion principle.

Lemma 2.5.12. *Given two points $y_{\mathtt{tt}}, y_{\mathtt{ff}} : Y$ and $p : y_{\mathtt{tt}} = y_{\mathtt{ff}}$, we get a map $f : I \longrightarrow Y$ satisfying the judgmental computation rules $f(|\mathtt{tt}|) \equiv y_{\mathtt{tt}}$ and $f(|\mathtt{ff}|) \equiv y_{\mathtt{ff}}$.*

Proof. The singleton type $(\Sigma y : Y) y_{\mathtt{tt}} = y$ is a proposition by Lemma 2.5.10, and has elements $(y_{\mathtt{tt}}, \mathsf{refl}(y_{\mathtt{tt}}))$ and $(y_{\mathtt{ff}}, p)$. Hence, by applying the recursion principle of propositional truncation, and the recursion principle of the Booleans, we get a map $f' : I \longrightarrow (\Sigma y : Y) y_{\mathtt{tt}} = y$ satisfying $f'(|\mathtt{tt}|) = (y_{\mathtt{tt}}, \mathsf{refl}(y_{\mathtt{tt}}))$ and $f'(|\mathtt{ff}|) = (y_{\mathtt{ff}}, p)$. Then composing f' with a projection map that forgets the equality gives the required map f. □

Lemma 2.5.13. *For two functions $f, g : X \longrightarrow Y$, given $p : (\Pi x : X) f(x) = g(x)$, we get $f = g$.*

Proof. Define $h' : X \longrightarrow I \longrightarrow Y$ by, for a given $x : X$, applying the recursion principle for I to $f(x)$, $g(x)$ and $p(x) : f(x) = g(x)$. Note that for a given x, we have that $h'(x)$ satisfies the judgmental equalities $h'(x, |\mathtt{tt}|) \equiv f(x)$ and $h'(x, |\mathtt{ff}|) \equiv g(x)$. This means that $h : I \longrightarrow X \longrightarrow Y$ defined by $h(i, x) :\equiv h'(x, i)$ satisfies $h(|\mathtt{tt}|) \equiv f$ and $h(|\mathtt{ff}|) \equiv g$.

Because $\|2\|$ is a proposition, we have $|\mathtt{tt}| = |\mathtt{ff}|$, so that by applying ap we get $f = g$. □

2.5.4 Univalence

In the previous sections we have paid attention to contractibility, propositions and sets. These notions are part of the hierarchy of *h-levels*, as is developed in The Univalent Foundations Program [91, Chapter 7], where the 'h' stands for *homotopy*. H-levels are phrased in terms of (iterated) identity types. Namely, for a fixed type X, two points $x, y : X$ give rise to the identity type $x =_X y$. Then, any two points $p, q : x =_X y$ of that type give rise to the identity type $p =_{x=_X y} q$. By repeating this process of building identity types of increasing nesting, we get the notion of iterated identity types. We then get a curious relationship between h-levels and identity types: a proposition X has, for any $x, y : X$, *contractible* identity types $x =_X y$. A set is a type X for which all types $x =_X y$ are propositions, and so given any $p, q : x =_X y$, the type $p =_{x=_X y} q$ is contractible. Such relationships can be developed in more generality.

The notion of sets arises unprompted from the notion of decidable equality in the following sense.

Lemma 2.5.14. *Consider a type $X : \mathcal{U}$. If X has decidable equality, that is, given an element of*

$$(\Pi x, y : X)(x = y) + \neg(x = y),$$

then X is a set.

This lemma is due to Hedberg [50], and is presented in more modern notation in Kraus et al. [62].

Another natural property of h-levels is that every contractible type is a proposition, and that every proposition is a set, and more generally h-levels are cumulative in a way reminiscent of homotopy theory. This natural role of h-levels has prompted some to develop homotopy-theoretic results in type-theoretic language [11, 95], so that, for instance, contractible homotopy types are modeled by contractible types. An appropriate notion modeling spacial equivalences is then the following type-theoretic notion.

Definition 2.5.15. For $X, Y : \mathcal{U}$ an *equivalence* is a map $f : X \to Y$ for which the following holds:

$$((\Sigma g : Y \to X)(\Pi x : X) g(f(x)) =_X x) \times ((\Sigma h : Y \to X)(\Pi y : Y) f(h(y)) =_Y y)\,.$$

We have the Σ-type $X \simeq Y$ consisting of maps $f : X \to Y$ that are equivalences in the above sense.

As is developed in The Univalent Foundations Program [91, Chapter 4], the above condition for $f : X \to Y$ is in fact a proposition.

In other words, a map $f : X \to Y$ is an equivalence if it has a left inverse g and a right inverse h. In particular, every isomorphism is an equivalence, since we can take the both-sided inverse $g : Y \to X$ for both single-sided inverses. Conversely, every equivalence $f : X \to Y$ is also an isomorphism [91, Chapter 4] in the sense that it has a both-sided inverse $g : Y \to X$. However, perhaps surprisingly, being an isomorphism is not a proposition [91, Theorem 4.1.3], so that the notion of equivalence is preferred, although a form of univalence phrased with isomorphisms has been considered in Hofmann and Streicher [53].

Equivalently, we may define $f : X \to Y$ to be an equivalence when for every $y : Y$, the *fiber* $(\Sigma x : X) f(x) = y$ is contractible. Recalling from Section 2.5.2 that contractibility can be read as "existence and uniqueness", this formalization can be read as stating that every fiber has a unique point.

An equivalence $f : X \to Y$, just like any map, induces a function $\mathrm{ap}_{f,x,x'} : x =_X x' \to f(x) =_Y f(x')$ on identity types. The fact that f is an equivalence implies that $\mathrm{ap}_{f,x,x'}$ is an equivalence for every $x, x' : X$. For the same reason, we get an equivalence on all iterated identity types, evocative of weak equivalences in homotopy theory.

Homotopically equivalent spaces are often considered equal in the same sense that isomorphic groups are considered equal. This leads us to consider an extensionality principle stating that an equivalence between types should correspond to an identification of types as elements of the universe $\mathcal{U}$.

This is made precise in a fashion similar to function extensionality. The direction given by dependent type theory is the following.

Lemma 2.5.16. *Given $X, Y : \mathcal{U}$ and $p : X = Y$, we get an equivalence $X \simeq Y$.*

Proof. By the induction principle of identity types, we may assume Y to be judgmentally equal to X, and the path to be reflexivity, so that we may give the identity function $\mathrm{id}_X : X \to X$ as an equivalence from X to X. □

We then obtain a converse to this lemma by requiring the construction to be an equivalence.

Definition 2.5.17. A universe $\mathcal{U}$ is *univalent* if the construction of Lemma 2.5.16 is itself an equivalence. In particular, this means that from an equivalence $X \simeq Y$ we obtain an identity $X = Y$.

In univalent type theory, it is assumed that every universe is univalent.

Axiom 2.5.18. Every universe is univalent.

Semantic justification is given by simplicial [58] and cubical [33, 78, 32] sets. Syntactic justification is given by 2-dimensional [67] and (Cartesian) cubical type theories [33, 5, 96].

We believe that the majority of our work goes through without full univalence, instead using the weaker extensionality principles discussed in the next section, although we have not checked this.

2.5.5 Consequences of extensionality

In Section 2.5.3, we obtained naive function extensionality by exploiting the computation rules of propositional truncations. Perhaps surprisingly, function extensionality can also be shown assuming univalence, as in The Univalent Foundations Program [91, Section 4.9] or Knapp [60, Corollary 2.45].

The following two consequences of univalence are often considered independently of univalence.

Definition 2.5.19. *Propositional univalence* is univalence for types $X, Y : \mathcal{U}$ when X and Y are known to be propositions. Explicitly, propositional univalence holds for $\mathcal{U}$ if for all types $X, Y : \mathcal{U}$ with $\mathrm{isHProp}(X)$ and $\mathrm{isHProp}(Y)$, the type of equivalences $X \simeq Y$ is equivalent to the type $X = Y$, that is,

$$(\Pi X, Y : \mathcal{U})\, \mathrm{isHProp}(X) \to \mathrm{isHProp}(Y) \to (X \simeq Y) \simeq (X = Y).$$

Definition 2.5.20. *Propositional extensionality*, another extensionality for propositions $X, Y : \mathcal{U}$, states than for all types $X, Y : \mathcal{U}$ with $\mathrm{isHProp}(X)$ and $\mathrm{isHProp}(Y)$, the type of pairs of maps $X \to Y$ and $Y \to X$ is equivalent to the type $X = Y$, that is,

$$(\Pi X, Y : \mathcal{U})\,\mathrm{isHProp}(X) \to \mathrm{isHProp}(Y) \to (X \Leftrightarrow Y) \simeq (X = Y).$$

We will often implicitly appeal to function extensionality to show that certain types are propositions.

Lemma 2.5.21. *Given $X : \mathcal{U}$ and $B : X \to \mathcal{U}$, if we have $p : (\Pi x : X)\,\mathrm{isHProp}(B(x))$, then* $\mathrm{isHProp}((\Pi x : X) B(x))$.

Proof. Consider $f, g : (\Pi x : X) B(x)$. In order to show equality of f and g, by function extensionality, it suffices to show equality pointwise. But $f(x) = g(x)$ by $p(x)$. □

This lemma finishes Discussion 2.4.6, where we needed that if Q is a proposition, then a type $\neg P \to Q$ is a proposition.

In summary, there is a variety of extensionality principles one may consider in the context of type theory, even for a focused area such as the two extensionality axioms for propositions above. UTT gives identity types a central role, regardless of whether we have opted to use certain extensionality principles.

2.6 Subtypes and embeddings

The extensionality discussed in the previous section means that a naive formalization of well-known subcollections, such as the even numbers as a subcollection of the naturals, will not behave as expected. Whereas in a material set theory, the collection of even numbers is distinct from the collection of all natural numbers, univalence would make the corresponding types identical, as we discussed in Section 2.4.1.

To avoid this unintended identification of the formalization of certain subcollections, we adjust the way we formalize subcollections in this section.

We will often be more explicit about universe levels, by amending Definition 2.4.1 so that for every universe level i, we have a type HProp_i of propositions in $\mathcal{U}_i$, instead of having a single symbol HProp for the type of propositions in an implicitly chosen universe $\mathcal{U}_i$.

Definition 2.6.1. By a *j-subtype* $B : \mathcal{P}_j A$ of $A : \mathcal{U}_i$ we mean a map $B : A \to \mathrm{HProp}_j$. For $b : A$ we define $(b \in B) :\equiv B(b)$. A *subtype* $B : \mathcal{P}A$ is a j-subtype $B : \mathcal{P}_j A$ for some universe level j. A *subset* is a subtype of a type that is a set in the sense of Definition 2.5.4.

This is motivated by the fact that if $B : \mathcal{P}_j A$ is a a subtype of A, then the projection map $\mathrm{pr}_0 : (\Sigma a : A)B(a) \to A$ is an *embedding*, and vice versa embeddings give rise to subtypes, as we will make precise in Lemma 2.6.3.

Definition 2.6.2. Given a function $f : C \to A$, we say f is an *embedding*, and write $f : C \hookrightarrow A$, if one of these two equivalent conditions holds:

$$(\Pi a : A)\ \mathrm{isHProp}\,((\Sigma c : C) f c = a) \tag{2.1}$$

$$(\Pi c, c' : C)\ \mathrm{isEquiv}(\mathrm{ap}_{f,c,c'}) \tag{2.2}$$

The second condition expresses that $\mathrm{ap}_{f,c,c'} : (c =_C c') \to (fc =_A fc')$ is an equivalence for all $c, c' : C$.

The fact that conditions (2.1) and (2.2) above are equivalent is a special case of Lemma 7.6.2 in The Univalent Foundations Program [91].

We use the notion of embedding, rather than that of injectivity, namely the requirement that $\mathrm{ap}_{f,c,c'}$ has a converse, for the convenient reason that "embedding + surjection = equivalence". If C and A are *sets*, then the identity types are propositions and hence embeddings coincide with injections.

Definitions 2.6.1 and 2.6.2 are equivalent in the following sense.

Lemma 2.6.3. *A subtype $B : \mathcal{P}_j A$ of $A : \mathcal{U}_i$ gives rise to a type $C : \mathcal{U}_{i \sqcup j}$ that embeds into A, where $i \sqcup j$ is the least universe level above i and j. Conversely, a type $C : \mathcal{U}_j$ with an embedding into $A : \mathcal{U}_i$ gives rise to a subtype $B : \mathcal{P}_{i \sqcup j} A$. These constructions are inverse to each other.*

Proof. In one direction, the type $C :\equiv (\Sigma b : A) b \in B$ embeds into A by the projection map. Conversely, given an embedding $f : C \hookrightarrow A$, the subtype is given by $B(a) :\equiv (\Sigma c : C)(fc = a)$, which is well-defined by the fact that f is an embedding. For details, see e.g. Rijke and Spitters [82, Theorem 2.29]. □

Remark 2.6.4. This result uses univalence to show an equality of types. It may be possible that Theorem 4.5.13, which uses it, can be shown without univalence.

We will often use this correspondence implicitly.

Lemma 2.6.5. *For types $A : \mathcal{U}_i$ and $B : \mathcal{U}_j$, and a relation $\leq : B \to B \to \mathrm{HProp}_j$, such that $(B, \leq)$ is a partial order, the type $A \to B$ with the ordering*

$$f \leq g :\equiv (\forall a : A) f(x) \leq g(x)$$

is a partial order. Explicitly, with $f, g, h : A \to B$:

1. *$f \leq f$,*
2. *$(f \leq g) \to (g \leq f) \to f = g$,*
3. *$(f \leq g) \to (g \leq h) \to (f \leq h)$.*

Proof. Straightforward, where antisymmetry uses function extensionality. □

Lemma 2.6.6. *The type HProp_j forms a partial order with the relation $\Rightarrow$. Explicitly, with $P, Q, R : \mathrm{HProp}_j$:*

1. *$P \Rightarrow P$,*
2. *$(P \Rightarrow Q) \to (Q \Rightarrow P) \to P = Q$,*
3. *$(P \Rightarrow Q) \to (Q \Rightarrow R) \to (P \Rightarrow R)$.*

Proof. Straightforward, where antisymmetry is propositional extensionality. □

Definition 2.6.7. For $A : \mathcal{U}_i$, $B : \mathcal{P}_j A$ and $C : \mathcal{P}_k A$, we define the ordering relation $\subseteq : \mathcal{P}_j A \to \mathcal{P}_k A \to \mathrm{HProp}_{i \sqcup j \sqcup k}$ by

$$P \subseteq Q :\equiv (\forall a : A)(a \in P) \Rightarrow (a \in Q).$$

Corollary 2.6.8. *For $A : \mathcal{U}_i$, the type $\mathcal{P}_j A$ of j-subtypes of A is a partial order with $\subseteq : \mathcal{P}_j A \to \mathcal{P}_j A \to \mathrm{HProp}_{i \sqcup j}$.*

Section 5.1 uses the following formulation of the antisymmetry of Corollary 2.6.8. Note that for embeddings from C and D into A, the relation $C \subseteq D$ holds if we have a certain commutative triangle as below.

Lemma 2.6.9. *Suppose given a triangle of maps as follows.*

$$\begin{tikzcd}
C \arrow[rr, "f"] \arrow[ddr, hook, "i_C"'] & & D \arrow[ddl, hook', "i_D"] \\
& \bigcirc & \\
& A &
\end{tikzcd}$$

If i_C and i_D are embeddings, and the commutativity condition $i_D \circ f = i_C$ is satisfied, then f is an embedding.

Proof. We can show, by induction on identity types, that $\mathrm{ap}_{i_C} = \mathrm{ap}_{i_D} \circ \mathrm{ap}_f$. Then, by a two-out-of-three property for equivalences [91, Theorem 4.7.1], we get that ap_f is an equivalence, as required. □

Lemma 2.6.10. *Suppose given a triangle of maps as follows.*

$$\begin{tikzcd}
C \arrow[rr, shift left, "f"] \arrow[ddr, hook, "i_C"'] & & D \arrow[ll, shift left, "g"] \arrow[ddl, hook', "i_D"] \\
& \bigcirc & \\
& A &
\end{tikzcd}$$

If i_C and i_D are embeddings, and the commutativity conditions $i_D \circ f = i_C$ and $i_C \circ g = i_D$ are satisfied, then f and g are equivalences.

Proof. It suffices to show that $g \circ f = \mathrm{id}_D$ and $f \circ g = \mathrm{id}_C$.

By the fact that i_C and i_D are embeddings, and using function extensionality, this is equivalent to showing that $i_D \circ g \circ f = i_D$ and $i_C \circ f \circ g = i_C$. Both cases can be shown by applying commutativity of the triangle both ways round. Explicitly, $i_D \circ g \circ f = i_C \circ f = i_D$ and $i_C \circ f \circ g = i_D \circ g = i_C$. □

2.6.1 Lattice-like structure of HProp

The types HProp and $\mathcal{P}A$, seen as partial orders, seem to have some additional structure reminiscent of being complete lattices. However, the naive construction of least upper bounds gives an element of a different type, namely propositions in a different universe.

Lemma 2.6.11. *For a given k-subtype $E : \mathcal{P}_k\mathrm{HProp}_j$ of HProp_j, the object $\bigvee E : \mathrm{HProp}_{(j+1)\sqcup k}$ defined by*

$$\bigvee E :\equiv (\exists P : \mathrm{HProp}_j) P \in E \wedge P$$

is an upper bound of E in the sense that

$$(\forall P : \mathrm{HProp}_j) P \in E \Rightarrow (P \Rightarrow \bigvee E)$$

and is minimal in the sense that

$$(\forall Q : \mathrm{HProp}_j)((\forall P : \mathrm{HProp}_j) P \in E \Rightarrow (P \Rightarrow Q)) \Rightarrow (\bigvee E \Rightarrow Q).$$

Similarly, $\bigwedge E : \mathrm{HProp}_{(j+1)\sqcup k}$ defined by $\bigwedge E :\equiv (\forall P : \mathrm{HProp}_j) P \in E \Rightarrow P$ is a maximal lower bound.

Proof. For the claim that $\bigvee E$ is an upper bound, let $P : \mathrm{HProp}_j$ with $P \in E$, and suppose P holds. Then by definition we have that $\bigvee E$ holds, as required.

For minimality, suppose $Q : \mathrm{HProp}_j$ is another upper bound, and that $\bigvee E$ holds. By the universal property of truncation, we may assume to have $P : \mathrm{HProp}_j$ with $P \in E$ such that P holds, and so because Q is an upper bound of E, we have that Q holds, as required.

The proof for $\bigwedge E$ is similar. □

Importantly, we do not claim that HProp_j is a complete lattice, because the object $\bigvee E$ we have constructed is an element of $\mathrm{HProp}_{(j+1)\sqcup k}$ rather than HProp_j. We will work towards a solution after also stating what the above situation means for the partial order of subsets of a type.

Lemma 2.6.12. *For any $A : \mathcal{U}_i$, for a given k-subtype $E : \mathcal{P}_k\mathcal{P}_jA$ of $\mathcal{P}_jA$, the object $\bigcup E : \mathcal{P}_{(j+1)\sqcup k}A$ defined by*

$$\left(\bigcup E\right)(a) :\equiv (\exists B : \mathcal{P}_jA)B \in E \land a \in B$$

is an upper bound of E in the sense that

$$(\forall B : \mathcal{P}_jA)B \in E \Rightarrow (B \subseteq \bigcup E)$$

and is minimal in the sense that

$$(\forall C : \mathcal{P}_jA)((\forall B : \mathcal{P}_jA)B \in E \Rightarrow (B \subseteq C)) \Rightarrow (\bigcup E \subseteq C).$$

Similarly, $\bigcap E : \mathcal{P}_{(j+1)\sqcup k}A$ defined by $(\bigcap E)(a) :\equiv (\forall B : \mathcal{P}_jA)B \in E \Rightarrow a \in B$ is a maximal lower bound.

When we think of HProp_i as a collection of truth values, motivated by the subobject classifier Ω in toposes, we may consider the possibility that there is only one such collection.

Definition 2.6.13. *Propositional resizing* holds if for any two universe levels i, j, we have

$$\mathrm{HProp}_i \simeq \mathrm{HProp}_j.$$

Corollary 2.6.14. *Assuming propositional resizing,* HProp_i *is a complete lattice. That is, for every $E : \mathcal{P}_k\mathrm{HProp}_i$, the proposition $\bigvee E : \mathrm{HProp}_i$ defined using propositional resizing by $\bigvee E :\equiv (\exists P : \mathrm{HProp}_i)P \in E \land P$ is a join of E, and similarly $\bigwedge E :\equiv (\forall P : \mathrm{HProp}_i)P \in E \Rightarrow P$ is a meet of E.*

Conversely, the claim that every HProp_i is a complete lattice implies propositional resizing [38], but we will not be using this.

Corollary 2.6.15. *Assuming propositional resizing, for any $A : \mathcal{U}_i$, the type $\mathcal{P}_jA$ of j-subsets of A is a complete lattice. That is, for any collection of subsets $E : \mathcal{P}_k\mathcal{P}_jA$, the intersection $\bigcup E : \mathcal{P}_jA$ defined using propositional resizing by*

$$\left(\bigcup E\right)(a) :\equiv (\exists B : \mathcal{P}_jA)B \in E \land a \in B$$

is a join of E, and similarly the map $\bigcap E : \mathcal{P}_j A$ defined by

$$\left(\bigcap E\right)(a) :\equiv (\forall B : \mathcal{P}_j A) B \in E \Rightarrow a \in B$$

is a meet of E.

2.6.2 Quantification over subtypes

Given a subtype $B : \mathcal{P}A$ of A, we sometimes consider *only* the elements of A that happen to be in B. In other words, we consider the elements of the type $(\Sigma b : A) b \in B$ corresponding via Lemma 2.6.3 to B. We introduce the following notation.

Definition 2.6.16. For $A : \mathcal{U}$, $B : \mathcal{P}A$ and $C : ((\Sigma a : A) a \in B) \to \mathcal{U}$ and $D : ((\Sigma a : A) a \in B) \to \mathrm{HProp}$, we write

$$\begin{aligned}(\Pi b \in B) C(b) &:\equiv (\Pi b : A)(\Pi v : b \in B) C(b, v),\\ (\Sigma b \in B) C(b) &:\equiv (\Sigma b : A)(\Sigma v : b \in B) C(b, v),\\ (\forall b \in B) D(b) &:\equiv (\forall b : A)(\forall v : b \in B) D(b, v),\\ (\exists b \in B) D(b) &:\equiv (\exists b : A)(\exists v : b \in B) D(b, v).\end{aligned}$$

For $C : A \to \mathcal{U}$ and $D : A \to \mathrm{HProp}$, this simplifies to the notation

$$\begin{aligned}(\Pi b \in B) C(b) &:\equiv (\Pi b : A) b \in B \to C(b),\\ (\Sigma b \in B) C(b) &:\equiv (\Sigma b : A) b \in B \times C(b),\\ (\forall b \in B) D(b) &:\equiv (\forall b : A) b \in B \Rightarrow D(b),\\ (\exists b \in B) D(b) &:\equiv (\exists b : A) b \in B \wedge D(b).\end{aligned}$$

For $C : \mathcal{U}$, this further simplifies to the notation for function types

$$B \to C :\equiv (\Pi b : A) b \in B \to C.$$

Remark 2.6.17. A different way to read the above notations is using the correspondence of Lemma 2.6.3, so that, for instance,

$$(\Pi b \in B)C(b) :\equiv (\Pi t : (\Sigma b : A)b \in B)C(t).$$

It is straightforward to check that this type coincides with the above interpretation.

Remark 2.6.18. Following the correspondence of Lemma 2.6.3, for $B : \mathcal{P}A$ and $C : \mathcal{U}$, we read $C \to B$ as the type $C \to (\Sigma b : A)b \in B$.

2.6.3 Identifications in subtypes

Let $B : A \to \mathrm{HProp}$ be a subtype of A. When are two elements of B equal? Note that this can be interpreted in two ways: either that we ask when two elements $u, v : (\Sigma b : A)b \in B$ of the Σ-type are equal, or that we ask when two $b, c : A$ with $b, c \in B$ have $b =_A c$.

In fact, these two notions are equal.

Lemma 2.6.19. *For $u, v : (\Sigma b : A)b \in B$, we have*

$$u =_{(\Sigma b:A)b\in B} v \Leftrightarrow \mathrm{pr}_1(u) =_A \mathrm{pr}_1(v).$$

Proof. We have to show two implications. For the first, let $u, v : (\Sigma b : A)b \in B$ and $p : u = v$ be arbitrary. By the induction principle of identity types, we may assume $u \equiv v$ and $p \equiv \mathsf{refl}(u)$, so that it suffices to prove $\mathrm{pr}_1(u) =_A \mathrm{pr}_1(u)$, which holds by $\mathsf{refl}(\mathrm{pr}_1(u))$.

For the second implication, let $u, v : (\Sigma b : A)b \in B$ be arbitrary. By the induction principle of Σ-types, we may assume that $u \equiv (b, s)$ and $v \equiv (c, t)$ with $s : b \in B$ and $t : c \in B$.

Now let $q : b =_A c$. By the induction principle of identity types, we may assume $b \equiv c$ and $q \equiv \mathsf{refl}(b)$, so that $s, t : b \in B$. It suffices to prove $(b, s) = (b, t)$, which holds because B is valued in propositions, so that $s = t$, and hence using $\mathrm{ap}_{(b,-),s,t} : s = t \to (b, s) = (b, t)$ we get the required result. □

In Section 2.5.2 we discussed that contractibility is used to formalize unique existence of an element with a given property. We can now make this more precise.

Lemma 2.6.20. *For* $B : A \to \mathrm{HProp}$*, the following two claims are equivalent:*

1. $\mathrm{isHProp}((\Sigma b : A) b \in B)$*, and*

2. $(\Pi b, c : A) b \in B \Rightarrow c \in B \Rightarrow b = c$.

Proof. Assuming $\mathrm{isHProp}((\Sigma b : A) b \in B)$, for any $b, c : A$ with $s : b \in B$ and $t : c \in B$ we get $(b, s) = (c, t)$ and hence $b = c$.

Assume $(\Pi b, c : A) b \in B \Rightarrow c \in B \Rightarrow b = c$. Given $u, v : (\Sigma b : A) b \in B$, we get $\mathrm{pr}_1(u) = \mathrm{pr}_1(v)$ and hence by Lemma 2.6.19 we get $u = v$. □

Lemma 2.6.21. *For* $B : A \to \mathrm{HProp}$*, the following two claims are equivalent:*

1. $\mathrm{isContr}((\Sigma b : A) b \in B)$*, and*

2. $(\Sigma b : A) b \in B \wedge ((\Pi c : A) c \in B \Rightarrow b = c)$.

Proof. Assuming $\mathrm{isContr}((\Sigma b : A) b \in B)$, we get the center of contraction $u : (\Sigma b : A) b \in B$, and can hence set $b \equiv \mathrm{pr}_1(u)$. The required claim $(\Pi c : A) c \in B \Rightarrow b = c$ then follows from Lemma 2.6.20.

Assume $b : A$ with $s : b \in B$ and that $(\Pi c : A) c \in B \Rightarrow b = c$. We can set the center of contraction to be (b, s), and then $\mathrm{isHProp}((\Sigma b : A) b \in B)$ follows from Lemma 2.6.20: for $c, d : A$ with $c, d \in B$ we get $c = b = d$. □

> *Remark* 2.6.22. The second equivalent claim expresses that we have some element of A satisfying property B, but uniqueness is only expressed about the element of A. In contrast, the first variant in terms of contractibility additionally expresses a kind of uniqueness of the proof of the property itself. But, since this property is propositional, this does not add any information.

2.7 Case study: quotient types

Propositional truncations can be thought of as taking a type, and adding all possible identifications between elements, so that it becomes a proposition. In a similar fashion, we can think of quotient types as taking a type and adding *some* identifications. Because identifications are added *freely*, attention has to be paid to the resulting identity types: when the binary relation relates an element x in a set X to itself, we still want the identity type $x = x$ to be contractible.

A straightforward solution, though perhaps unsatisfactory, is to, in addition to the identifications coming from the relation, add further identifications that force the constructed quotient type to be a set. Quotients are constructed in more generality in Rijke [81], where such set-quotients fit in a hierarchy also including propositional truncation and Rezk completion [1], and where the resulting set-quotient is a set automatically, rather than it being forced by an additional constructor.

Definition 2.7.1. A *binary relation* on a type $X : \mathcal{U}$ means a map $R : X \to X \to \mathrm{HProp}$. The *quotient type* X/R is then defined as a higher-inductive type given by three constructors:

1. an inclusion constructor $i : X \to X/R$,
2. an equivalence class identification constructor $c : (\Pi x, y : X) R(x, y) \to i(x) = i(y)$, and
3. a set-squashing constructor $s : (\Pi s, t : X/R)(\Pi p, q : s = t) p = q$,

noting that the last constructor can also be read as, simply, $\mathrm{isSet}(X/R)$.

An *equivalence class* is an element of X/R, and the equivalence class of $x : X$ is simply $i(x)$. By abuse of terminology, we may say that $x : X$ is an *element of an equivalence class* $s : X/R$ when $s = i(x)$.

Axiom 2.7.2. For each X and R as above we have a quotient type X/R in our type theory.

Remark 2.7.3. In contrast to traditional set-theoretic constructions of quotients, we do not require that the binary relation R is an equivalence relation: its only role is to cause sufficiently many elements of X/R to be identified. Regardless, the identity types of X/R become equivalence relations for type-theoretic reasons. For instance, reflexivity is simply given by the refl constructor of identity types.

Considering the above constructors, Remark 2.3.5, and the recursion principle for propositional truncations in Definition 2.4.3, the recursion principle is as follows. For a codomain $B : \mathcal{U}$, such that B is a set, we obtain a map $f : X/R \to B$ from the following data:

1. a map $f_i : X \to B$, and

2. for each $x, y : X$ with $R(x, y)$, an element of $f_i(x) = f_i(y)$.

Then f satisfies the computation rule that for every $x : X$, we have $f(i(x)) \equiv f_i(x)$.

There is also a corresponding induction principle, which is normally stated in terms of dependent paths.

2.8 Classical principles

The majority of existing work in analysis has been developed in a classical framework, and for that reason we need to be able to relate results to principles used in classical mathematics.

Again, care has to be taken when formalizing statements.

Lemma 2.8.1. *Naive excluded middle*

$$(\Pi X : \mathcal{U}) X + \neg X$$

is false.

A proof can be found in The Univalent Foundations Program [91, Corollary 3.2.7]. The problem can be thought of as follows. An element $f : (\Pi X : \mathcal{U}) X + \neg X$ chooses a point of X, whenever this is possible. If we permute the elements of X, then f chooses the same point, contradicting the fact that f is equivariant under equivalences, as follows from univalence. So, concretely, suppose $f(\mathbf{2}) = \mathsf{tt}$. Using an identification $\mathbf{2} =_{\mathcal{U}} \mathbf{2}$, obtained via univalence from an equivalence swapping the two points of $\mathbf{2}$, we can show that $f(\mathbf{2}) = \mathsf{ff}$, so that $\mathsf{tt} = \mathsf{ff}$, a contradiction.

We can take away the need for f to choose a point by restricting to propositions.

Definition 2.8.2. The *principle of excluded middle* is the claim that every proposition is either true or false, that is:

$$\mathrm{PEM} :\equiv (\Pi P : \mathrm{HProp}) P + \neg P.$$

Discussion 2.8.3. The inclusion (or not) of the principle of excluded middle has a strong impact on the mathematics in a given foundational setting. It is often desirable to compare the provability of a given claim in a system without excluded middle with the provability of that claim in a system with excluded middle. The important situation is where a claim C is provable in a logic L+PEM that includes PEM, but we know that no proof exists in the weaker logic L, since we have a semantics of L where the negation $\neg C$ of the claim holds, and where thus PEM must be false. The terminology "C is not provable in L" refers to the meta-theoretical claim that there is no proof of C in L, usually because it is independent. This should be contrasted with the terminology "C is false in L", which has not been established in the described situation.

We will also consider following consequences of PEM. Recall the type DHProp of decidable propositions from Definition 2.4.9.

Definition 2.8.4.

1. The *limited principle of omniscience* is the claim that for every decidable predicate $P : \mathbb{N} \to \mathrm{DHProp}$ on the naturals, we can decide $(\exists n : \mathbb{N})P(n)$:

$$\mathrm{LPO} :\equiv (\Pi P : \mathbb{N} \to \mathrm{DHProp})((\exists n : \mathbb{N})P(n)) + \neg((\exists n : \mathbb{N})P(n)).$$

2. The *weak limited principle of omniscience* is the claim that for every decidable predicate $P : \mathbb{N} \to \mathrm{DHProp}$ on the naturals, we can decide $\neg(\exists n : \mathbb{N})P(n)$:

$$\mathrm{WLPO} :\equiv (\Pi P : \mathbb{N} \to \mathrm{DHProp})\neg((\exists n : \mathbb{N})P(n)) + \neg\neg((\exists n : \mathbb{N})P(n)).$$

Remark 2.8.5. As we will see in Theorem 6.5.1, we have that from $(\exists n : \mathbb{N})P(n)$ we get $(\Sigma n : \mathbb{N})P(n)$.

We can see WLPO as a consequence of *weak excluded middle*, which, for an arbitrary type X, decides $\neg X$:

$$\mathrm{WEM} :\equiv (\Pi X : \mathcal{U})\neg X + \neg\neg X.$$

In Lemma 6.10.2 we will prove WLPO from the existence of a strongly nonconstant function from the reals to the Booleans.

The principles PEM and WEM can be characterized by specific violations of parametricity [24]. See Bernardy, Jansson, and Paterson [16] and Atkey, Ghani, and Johann [7] for notions of parametricity extended to dependent type theory. For example, PEM is equivalent to the existence of a function $f : (\Pi X : \mathcal{U})X \to X$ and a type $X : \mathcal{U}$ with a point $x : X$ such that $f_X(x) \neq x$. If we have pushouts [81], then WEM is equivalent to the existence of a function $f : (\Pi X : \mathcal{U})X \to \mathbf{2}$ with types $X, Y : \mathcal{U}$ and elements $x : X$, $y : Y$ with $f_X(x) \neq f_Y(y)$.

We also consider choice-like axioms. For similar reasons as above, we have to make some restrictions to the types involved to avoid inconsistencies.

Definition 2.8.6.

1. The *axiom of choice* states that, for every *set* $X : \mathcal{U}$, and type family $Y : X \to \mathcal{U}$ such that $Y(x)$ is a set for each $x : X$, we have the implication

$$((\Pi x : X) \, \|Y(x)\|) \to \|(\Pi x : X) Y(x)\| \, .$$

2. *Countable choice* is the axiom of choice with $\mathbb{N}$ for X. That is, for every type family $Y : \mathbb{N} \to \mathcal{U}$ such that $Y(n)$ is a set for each $n : \mathbb{N}$, we have the implication

$$((\Pi n : \mathbb{N}) \, \|Y(n)\|) \to \|(\Pi n : \mathbb{N}) Y(n)\| \, .$$

2.9 Notes

We have presented parts of the type theory introduced in The Univalent Foundations Program [91], and have included some attention for the proof-theoretic underpinnings in our discussion. This will allow us to make a connection with proof assistants in Section 9.1.

Section 2.5 described an important distinction between general type theories and UTT. We have described function extensionality to some extent, with a focus on independent functions, to avoid considering homotopy-theoretic concepts such as dependent identifications. Contractibility of singletons in Lemma 2.5.10 is traditionally shown in terms of such dependent identifications, which we have avoided for the sake of more elementary arguments.

Although subtypes are considered in The Univalent Foundations Program [91], we have expanded on it somewhat: we have paid more attention to universe levels, and have introduced a notation for quantification in subtypes in Definition 2.6.16. This allows us to make sense of expressions such as $(\forall x \in E)\phi(x)$, where E is the subset of the naturals consisting of the even numbers. This may falsely lead some readers to believe that we are using set-theoretic, rather than type-theoretic, foundations.

Chapter 3

FIXPOINTS IN DCPOS

In Section 2.6 we discussed the partial order $\mathcal{P}A$ of subtypes of a fixed type A. Assuming propositional resizing, this partial order has joins and meets.

In this chapter we develop the theory of directed-complete partial orders (dcpo), namely partially ordered sets in which only certain joins are required to exist. We apply this theory in Section 5.2 to show that two types of real numbers coincide by computing a certain fixpoint in a dcpo.

The theory of dcpos is well understood in toposes, where we have a single object Ω of truth values, and a single notion of subset.

In univalent type theory, we have a hierarchy of truth values HProp_i indexed by universe levels i. Correspondingly, we also have a powerset $\mathcal{P}_iA$ for each universe level i, and although we can upgrade elements of $\mathcal{P}_iA$ to elements of $\mathcal{P}_{i+1}A$, the various $\mathcal{P}_iA$ are not automatically equivalent. Consequently, there are choices to be made when defining dcpos. This is why we develop the theory of dcpos in some detail in this chapter, paying extra attention to universe levels. This chapter can be seen as a simplification of the general approach of de Jong [57]: we have taken some universe levels to be equal.

We assume propositional resizing to construct the dcpo in our application in Section 5.2, so that, as in Section 2.6, all joins exist in $\mathcal{P}_iA$. Additionally, once we have propositional resizing, $\mathcal{P}_iA$ is equivalent to $\mathcal{P}_kA$ for all i, k, so that one may expect the topos-theoretic approach may work out as usual. The purpose of this chapter is to investigate if propositional resizing suffices to resolve all issues with the hierarchy of truth values, and where exactly propositional resizing is used in fixpoint theorems.

Note that throughout this chapter, we make extensive usage of the notation for subsets as in Section 2.6.

3.1 Dcpos

We start by defining partial orders. By a binary relation R on a set X, we mean a map $X \to X \to \mathrm{HProp}$, as in Definition 2.7.1. We can specify which universe the binary relation lands in by saying that R is HProp_i*-valued* or is a *relation in universe* i if $R : X \to X \to \mathrm{HProp}_i$.

Definition 3.1.1. A binary relation R on a set X is

1. *reflexive* if $(\forall x : X) Rxx$;
2. *irreflexive* if $(\forall x : X) \neg Rxx$;
3. *symmetric* if $(\forall x, y : X) Rxy \Rightarrow Ryx$;
4. *antisymmetric* if $(\forall x, y : X) Rxy \Rightarrow Ryx \Rightarrow x = y$;
5. *transitive* if $(\forall x, y, z : X) Rxy \Rightarrow Ryz \Rightarrow Rxz$;
6. *cotransitive* if $(\forall x, y, z : X) Rxy \Rightarrow (Rxz) \vee (Rzy)$.

> *Remark* 3.1.2. Cotransitivity is also known as *weak linearity* [91, Definition 11.2.7] or the *approximate splitting principle* [84].

Definition 3.1.3.

A *preorder*, denoted by $\leq$, is a reflexive transitive relation.

A *partial order* is an antisymmetric preorder.

Definition 3.1.4. Let $(A, \leq)$ be a partially ordered set, as in Definition 3.1.3.

1. An endomap $f : A \to A$ is *inflationary* if it is *monotonic*, i.e. $(\forall x, y : A) x \leq y \Rightarrow f(x) \leq f(y)$, and *increasing*, i.e. $(\forall x : A) x \leq f(x)$.

2. A subset $\mathcal{D} : \mathcal{P}A$ of A is *semidirected* if whenever $x, y \in \mathcal{D}$, there exists $z \in \mathcal{D}$ with $x \leq z$ and $y \leq z$.

3. A subset $\mathcal{D}$ of A is *directed* if it is semidirected and inhabited.

4. $(A, \leq)$ is a *directed-complete partial order (dcpo)* if every directed subset $\mathcal{D}$ of A has a join in A, i.e. has an upper bound $w : A$ of $\mathcal{D}$ such that if v is also an upper bound of $\mathcal{D}$, then $w \leq v$.

5. A subset $X : \mathcal{P}A$ of a dcpo $(A, \leq)$ is a *subdcpo* if whenever $\mathcal{D}$ is a directed subset of A contained in X, its join is contained in X.

6. We can be more explicit about universe levels, discussed in Section 2.3.1, by saying, when the ordering relation on some $A : \mathcal{U}_i$ is valued in HProp_i, that $(A, \leq)$ is an *i-dcpo* if every directed subset $\mathcal{D} : \mathcal{P}_i A$ has a join in A.

7. Similarly, a subset $X : \mathcal{P}_i A$ of an i-dcpo $(A, \leq)$ is an *i-subdcpo* if for every directed subset $\mathcal{D} : \mathcal{P}_i A$ with $\mathcal{D} \subseteq X$, its join in the dcpo $(A, \leq)$ is an element of X.

The following lemma justifies the name subdcpo. There is also a converse formulation, which we will not use.

Lemma 3.1.5. *An i-subdcpo $B : \mathcal{P}_i A$ of an i-dcpo $(A, \leq)$ gives rise to an i-dcpo $((\Sigma b : A) b \in B, \leq)$ of elements in B with the ordering given by restriction as*

$$(b, \mu) \leq (b', \mu') :\equiv b \leq b'.$$

Proof. Let $\mathcal{D} : \mathcal{P}_i((\Sigma b : A) b \in B)$ be a directed subset of $(\Sigma b : A) b \in B$. Then the subset $\mathcal{D}' : \mathcal{P}_i A$ defined by

$$(d \in \mathcal{D}') :\equiv (\Sigma \mu : d \in B)((d, \mu) \in \mathcal{D})$$

is directed because $\mathcal{D}$ is, and contained in the subdcpo B, so that it has a join in B, that is, $\bigvee \mathcal{D}' \in B$, so that it has a join in $(\Sigma b : A) b \in B$, as required. □

Given any dcpo, the function type with that dcpo as its codomain is again a dcpo.

Lemma 3.1.6. *Given a type $A : \mathcal{U}_i$ and an i-dcpo $(B, \leq)$, the type $A \to B$ with the pointwise ordering $f \leq g :\equiv (\forall x : A) f(x) \leq g(x)$ is an i-dcpo.*

Proof. The given relation is a partial order by Lemma 2.6.5.

Let $\mathcal{D} : \mathcal{P}_i(A \to B)$ be directed. We construct a join $d : A \to B$ of $\mathcal{D}$ pointwise. For an arbitrary $x : A$, we claim the subset $\mathcal{D}[x] : \mathcal{P}_i B$ of B defined by

$$(y \in \mathcal{D}[x]) :\equiv (\exists f : A \to B) f \in \mathcal{D} \wedge y = f(x)$$

is directed. It is inhabited because $\mathcal{D}$ is. For semidirectedness, consider $y, z \in \mathcal{D}[x]$, so that there exist $f, g : A \to B$ with $f, g \in \mathcal{D}$ and $y = f(x)$ and $z = g(x)$. Then since $f, g \in \mathcal{D}$, there exists $h \in \mathcal{D}$ with $f, g \leq h$, and so because of the pointwise ordering on $A \to B$ we have $f(x), g(x) \leq h(x)$, i.e. $y, z \leq h(x)$, as required.

Hence $\mathcal{D}[x]$ has a join that we call $d(x)$, so that $d : A \to B$, and d is an upper bound of $\mathcal{D}$ because $\mathcal{D}[x] \leq d(x)$ for each x. If $e : A \to B$ is another upper bound of $\mathcal{D}$, then for each $x : A$ we have $\mathcal{D}[x] \leq d(x) \leq e(x)$, and hence $d \leq e$ because the ordering is pointwise, so that d is indeed the least upper bound of $\mathcal{D}$. □

Finally, we have the main example of a dcpo, which will also be used in Section 5.2.

Lemma 3.1.7. *Assuming propositional resizing, for any $A : \mathcal{U}_i$, the type $\mathcal{P}_i A$ of subtypes of A is an $(i+1)$-dcpo under the $\subseteq$ ordering.*

Proof. By Lemma 2.6.8, $\subseteq : \mathcal{P}_i A \to \mathcal{P}_i A \to \mathrm{HProp}_i$ is a partial order.

We can form the join of $\mathcal{D} : \mathcal{P}_{(i+1)} \mathcal{P}_i A$ using Corollary 2.6.15. □

3.2 Fixpoints

Given a certain endomap $f : A \to A$ on a dcpo, we aim to construct a fixpoint of f. Pataraia's fixed point theorem, in Theorem 3.2.3 below, which merely assumes f to be monotonic, computes a fixpoint using propositional resizing. Perhaps surprisingly, if we additionally have that f is increasing, so that it is inflationary, then we do not need propositional resizing, and this is Corollary 3.2.2 below.

Proposition 3.2.1 (Pataraia [79], Escardó–Simpson [43]). *Let $(A, \leq)$ be an i-dcpo. The subset $I : \mathcal{P}_i(A \to A)$ of $A \to A$ of inflationary endomaps, given by*

$$(f \in I) :\equiv ((\forall x, y : A) x \leq y \Rightarrow f(x) \leq f(y)) \wedge ((\forall x : A) x \leq f(x)),$$

is an i-subdcpo. I is a directed subset of I, so that I has a top element $\top$. Given a point $x : A$, $\top(x)$ is a common fixed point of all inflationary maps on A.

Proof. Let $\mathcal{D} \subseteq I$ be directed. To show that its join $\bigvee \mathcal{D}$ in $A \to A$ is an inflationary map, notice that if $x \leq y$ in A then $(\bigvee \mathcal{D})(y)$ is an upper bound of $\mathcal{D}[x]$, and that for $x : A$ and any $f \in \mathcal{D}$, we have $x \leq f(x)$, so that $(\bigvee \mathcal{D})(x)$ is an upper bound of $f(x)$ and hence of x.

I is semidirected in I because for $f, g \in I$ we have $f, g \leq f \circ g$, and inhabited because the identity map is inflationary. Hence I is directed.

Let $x : A$ and let $f : A \to A$ be inflationary, so that in particular $\top \leq f \circ \top$. Since $f \in I$, we have $f \circ \top \leq \top$, and hence $f \circ \top = \top$, making $\top(x)$ a fixed point of f. □

The following corollary is the fixed point theorem we will use in Section 5.2.

Corollary 3.2.2. *Let $(A, \leq)$ be an i-dcpo, and $f : A \to A$ an inflationary endomap. If $B : \mathcal{P}_i A$ is an f-closed i-subdcpo of A, then from a point of B we can construct a fixed point of f.*

Proof. The type $(\Sigma b : A) b \in B$ in $\mathcal{U}_i$ of elements in B is an i-dcpo by Lemma 3.1.5, and $f : A \to A$ gives rise to an inflationary endomap on it, so that Proposition 3.2.1 applies. □

For completeness, we include a type-theoretic proof of Pataraia's fixed point theorem, which we will not use. Note the usage of propositional resizing for the construction of an intersection B' of subdcpos.

Theorem 3.2.3 (Pataraia's fixed point theorem [43, 79]). *Let $(A, \leq)$ be an i-dcpo with a bottom element, and let $f : A \to A$ be a monotonic endomap. Assuming propositional resizing, f has a least fixed point.*

Proof. Using propositional resizing, we can define the intersection $B' : \mathcal{P}_i A$ of all i-subdcpos of A that contain the least element $\perp$ of A and that are closed under f as in Corollary 2.6.15:

$$(x \in B') :\equiv (\forall B : \mathcal{P}_i A)[\mathrm{subdcpo}_i(B) \wedge \perp \in B \wedge ((\forall y \in B) f(y) \in B)] \Rightarrow x \in B,$$

where $\mathrm{subdcpo}_i(B)$ expresses that B is an i-subdcpo of A. Now B' is an f-closed i-subdcpo. Additionally, f is an inflationary map on B' because if $x \in B'$ then x is also an element of the i-subdcpo given by $B(x) :\equiv x \leq f(x)$, and hence $x \leq f(x)$.

By Corollary 3.2.2, f has a fixed point $a \in B'$. It remains to show that a is the least fixed point of f, which we do by showing the more general property that it is the least prefixed point of f. So let y be a prefixed point of f, that is, $f(y) \leq y$. Then the down set of y is an i-subdcpo, which is f-closed and contains $\bot$, and hence also contains a, so that $a \leq y$. □

By analysing in detail how propositional resizing is used in the basic theory of dcpos and Pataraia's fixpoint theorem, we have formulated a fixpoint theorem that does not use propositional resizing. Our proof in Section 5.2 only uses propositional resizing in order to appeal to Lemma 3.1.7.

3.3 Notes

The work in this chapter was partially developed contemporaneously with de Jong [57], which we recommend for the general treatment of dcpos in UTT. Our definition of dcpos, and our version of Pataraia's theorem, were written solely to be applied in the proof of Theorem 5.2.1.

Chapter 4

REAL NUMBERS

In order to do analysis, we need a solid understanding of real numbers. Traditionally, analysis is developed either using an axiomatic collection of real numbers, such as an arbitrary Cauchy complete Archimedean field, or using a concrete set of real numbers, for example a set of Dedekind cuts. Notions from analysis such as continuity, trigonometric functions, and integration, can then be developed in terms of the chosen real numbers.

Our development of constructive analysis and metric spaces in Chapters 6, 7 and 8 uses the former approach of assuming an arbitrary type of reals. This chapter presents the appropriate notion of Cauchy complete Archimedean field, and presents a number of concrete sets of real numbers, thereby justifying the axiomatic approach.

Definitions that are classically well-understood require re-examining in a constructive setting. Already the definition of a field is problematic, as we discuss in Section 4.1, because the assumption that a number is not equal to 0 is often not strong enough to allow the computation of a multiplicative inverse. We solve this, as is usual in constructive mathematics, by using an apartness relation instead of the negation of equality, so that inverses may be computed from the assumption that a number is *apart* from 0.

In the specific case of real numbers, this apartness relation arises from the ordering as $x \# y \coloneqq (x < y) \vee (y < x)$, and so the notion of *constructive field*, which is equipped with an arbitrary apartness relation, specializes to *ordered fields* in which the apartness relation arises from the strict order as described.

We state the Archimedean condition in the style of the The Univalent Foundations Program

[91] in Section 4.3, where we try to be more explicit about the role of the rational numbers.

The concrete sets of reals also deserve attention. For example, the classically equivalent sets of Cauchy reals and Dedekind reals cannot be shown to be equivalent in our constructive setting, although we do have the usual canonical inclusion from the Cauchy reals into the Dedekind reals. Additionally, highly surprisingly from a classical standpoint, the Cauchy reals, namely equivalence classes of Cauchy sequences in the rationals, cannot be shown to be Cauchy complete, as we discuss in Section 4.4. This then leads us to consider the HoTT book reals in Section 4.5, which is a Cauchy complete Cauchy completion of the rationals. Another possibility is given by the Dedekind reals in Section 4.6, which are easier to understand from a type-theoretic point of view, but cannot live in the lowest universe.

During formalization, attention should be paid to what should be property and what should be structure, so that, for example, we avoid duplicating Dedekind reals because we accidentally collected cut-bound pairs, in which a single real number is represented several times with different upper and lower bounds, rather than Dedekind cuts with the *property* of boundedness. As a rule of thumb, constructions of sets correspond to *structures* in type theory, while logical claims correspond to *properties* formalized using univalent logic as in Definition 2.4.5, where we use the emphasized terms as in Definition 2.4.7. One important reason to deviate from this rule of thumb is for considerations of constructivity, for example our focus on Cauchy sequences with modulus as in Definition 4.4.1.

Univalent type theory allows us to define sets of real numbers, with identity types directly capturing the intended equality of real numbers. By contrast, in setoid-based approaches such as in e.g. Bishop and Bridges [20] and Bauer et al. [12], an equivalence relation is specified capturing the intended equality. Whereas in setoid-based approaches, there is an *understanding* that the map $x, y \mapsto x + y$ has to be checked to be invariant under the equivalence relation, in a univalent approach this requirement is automatically *enforced* by the type theory, so that the only definable maps are those that preserve equality on the reals.

The definitions and results in this chapter can essentially be found in existing literature. We rephrase them here to make the relation between property and structure more explicit. As an example, consider the exponential function $\exp(x) = \sum_{k=0}^{\infty} \frac{x^k}{k!}$ which computes its output as the limit of a series. If we can only compute limits for sequences with a *modulus* of Cauchy convergence, we seem to need to have such a modulus as a *structure*, whose construction requires discrete data about the input x. We avoid requiring this data by observing that the type of limits of a fixed sequence is a proposition, so that the *property* of existence of a modulus for the series suffices to compute limits. As another example, in Chapter 6 we will rephrase a certain locatedness property into structure, allowing us to compute with real numbers in an algebraic way, while also being able to compute discrete results such as digit representations.

In summary, we define Cauchy complete Archimedean ordered fields in Sections 4.1–4.3, Cauchy reals in Section 4.4, the HoTT book reals in Section 4.5, and Dedekind cuts in Section 4.6.

4.1 Algebraic structure of numbers

In this section we give some definitions of fields. The most important notion is that of an *ordered field* in Definition 4.1.10, which in Sections 4.3 and 4.4 be augmented to a Cauchy complete Archimedean field.

First, we consider the classical definition of a field. We define apartness relations, ordering relations, and lattices, so that we can define constructive fields and ordered fields.

Fields have the property that nonzero numbers have a multiplicative inverse, or more precisely, that

$$(\forall x : F) x \neq 0 \Rightarrow (\exists y : F) x \cdot y = 1.$$

Remark 4.1.1. If we require the collection of numbers to form a set in the sense of Definition 2.5.4, and satisfy the ring axioms, then multiplicative inverses are unique, so that the above is equivalent to the proposition

$$(\Pi x : F) x \neq 0 \Rightarrow (\Sigma y : F) x \cdot y = 1.$$

Classically, existence of multiplicative inverses can be shown for the Cauchy reals by starting with a real number represented by an element of its equivalence class $x = [(x_n)_n]$, and inverting it pointwise to obtain $y = [(x_n^{-1})_n]$, taking 0^{-1} to be 1. By several applications of LPO, we can show that $(x_n^{-1})_n$ is a Cauchy sequence, and that y is a multiplicative inverse of x. We use LPO to obtain an index from which onward $(x_n)_n$ is bounded away from $\varepsilon > 0$.

Constructively, in the case of finite fields, rationals, and algebraic numbers, the assumption $x \neq 0$, where by $x \neq y$ we mean the negation $\neg(x = y)$ of equality, is strong enough to compute multiplicative inverses. These rings are fields in the classical sense.

Definition 4.1.2. A *classical field* is a set F with points $0, 1 : F$, operations $+, \cdot : F \longrightarrow F \longrightarrow F$, which is a commutative ring with unit, such that

$$(\forall x : F) x \neq 0 \Rightarrow (\exists y : F) x \cdot y = 1.$$

> *Remark* 4.1.3. As in the classical case, by proving that additive and multiplicative inverses are unique, we also obtain the negation and division operations.

For the reals, the assumption $x \neq 0$ does not give us any information allowing us to bound x away from 0, which we would like in order to compute multiplicative inverses.

Hence, we give a variation on the definition of fields in which the underlying set comes equipped with an *apartness relation* #, which satisfies $x \# y \Rightarrow x \neq y$, although the converse implication may not hold. This apartness relation allows us to make appropriate error bounds and compute multiplicative inverses based on the assumption $x \# 0$.

Definition 4.1.4.

An *apartness relation*, denoted by #, is an irreflexive symmetric cotransitive relation.

A *strict partial order*, denoted by <, is an irreflexive transitive cotransitive relation.

Definition 4.1.5. A *constructive field* is a set F with points $0, 1 : F$, binary operations $+, \cdot :$ $F \longrightarrow F \longrightarrow F$, and a binary relation # such that

1. $(F, 0, 1, +, \cdot)$ is a commutative ring with unit;

2. $x : F$ has a multiplicative inverse iff $x \# 0$;

3. # is tight, i.e. $\neg(x \# y) \Rightarrow x = y$;

4. + is #-extensional, that is, for all $w, x, y, z : F$

$$w + x \# y + z \Rightarrow w \# y \vee x \# z.$$

Lemma 4.1.6. *For a constructive field* $(F, 0, 1, +, \cdot, \#)$*, the following hold.*

1. $1 \# 0$.

2. Addition + is #-compatible in the sense that for all $x, y, z : F$

$$x \# y \Leftrightarrow x + z \# y + z.$$

3. Multiplication · is #-extensional in the sense that for all $w, x, y, z : F$

$$w \cdot x \# y \cdot z \Rightarrow w \# y \vee x \# z.$$

Proof. The first item follows because 1 has multiplicative inverse 1.

For #-compatibility of +, suppose $x \# y$, that is, $(x+z)-z \# (y+z)-z$. Then #-extensionality gives $(x + z \# y + z) \vee (-z \# -z)$, where the latter case is excluded by irreflexivity of #. The other direction is similar.

To show #-extensionality of ·, suppose $w \cdot x \# y \cdot z$. By cotransitivity of #, we get $(w \cdot x \# w \cdot z) \vee (w \cdot z \# y \cdot z)$. By #-compatibility of +, we have that $w \cdot x \# w \cdot z$ implies $0 \# w \cdot (z - x)$, so that $z - x$ has multiplicative inverse $w \cdot \frac{1}{w \cdot (z-x)}$, and hence $x \# z$. In the case $w \cdot z \# y \cdot z$, we get $0 \# z \cdot (y - w)$, so that $y - w$ has multiplicative inverse $z \cdot \frac{1}{z \cdot (y-w)}$, and hence $w \# y$. □

In the case of the real numbers, that we will develop in Sections 4.4–5.1, apartness # and the ordering relation ≤ arise from its strict partial order <, as follows.

Lemma 4.1.7. *Given a strict partial order < on a set X:*

1. we have an apartness relation defined by

$$x \# y :\equiv (x < y) \vee (y < x), \quad \textit{and}$$

2. *we have a preorder defined by*

$$x \leq y :\equiv \neg(y < x).$$

Proof. Straightforward. □

Whenever we have a strict partial order on a set X, by # and $\leq$ we mean the relations induced by Lemma 4.1.7.

We specialize the notion of constructive field to that of *ordered field* in which the strict partial order $<$ is suitably compatible with the algebraic operations. In addition, we want ordered fields to come equipped with joins and meets.

Definition 4.1.8. Let $(A, \leq)$ be a partial order, and let $\min, \max : A \to A \to A$ be binary operators on A. We say that $(A, \leq, \min, \max)$ is a *lattice* if min computes greatest lower bounds in the sense that for every $x, y, c : A$, we have

$$c \leq \min(x, y) \Leftrightarrow c \leq x \wedge c \leq y,$$

and max computes least upper bounds in the sense that for every $x, y, c : A$, we have

$$\max(x, y) \leq c \Leftrightarrow x \leq c \wedge y \leq c.$$

Remark 4.1.9.

1. From the fact that $(A, \leq, \min, \max)$ is a lattice, it does *not* follow that for every x and y, $\max(x, y) = x \vee \max(x, y) = y$, which would hold in a linear order. However, in Lemma 6.7.1 we characterize max as

$$z < \max(x, y) \Leftrightarrow z < x \vee z < y,$$

and similarly for min.

2. In a partial order, for two fixed elements a and b, all joins and meets of a, b are equal, so that Lemma 2.6.20 the type of joins and the type of meets are propositions. Hence, providing the maps min and max as in the above definition is equivalent to

the showing the *existence* of all binary joins and meets.

The following definition is modified from on The Univalent Foundations Program [91, Definition 11.2.7].

Definition 4.1.10. An *ordered field* is a set F together with constants 0, 1, operations $+$, $\cdot$, min, max, and a binary relation $<$ such that:

1. $(F, 0, 1, +, \cdot)$ is a commutative ring with unit;

2. $<$ is a strict order;

3. $x : F$ has a multiplicative inverse iff $x \# 0$, recalling that $\#$ is defined as in Lemma 4.1.7;

4. $\leq$, as in Lemma 4.1.7, is antisymmetric, so that $(F, \leq)$ is a partial order;

5. $(F, \leq, \min, \max)$ is a lattice.

6. for all $x, y, z, w : F$:

$$x + y < z + w \Rightarrow x < z \vee y < w, \tag{†}$$

$$0 < z \wedge x < y \Rightarrow xz < yz. \tag{∗}$$

Our notion of ordered fields coincides with The Univalent Foundations Program [91, Definition 11.2.7].

Lemma 4.1.11. *In the presence of the first five axioms of Definition 4.1.10, conditions (†) and (∗) are together equivalent to the condition that for all $x, y, z : F$,*

1. $x \leq y \Leftrightarrow \neg(y < x)$,

2. $x \# y \Leftrightarrow (x < y) \vee (y < x)$,

3. $x \leq y \Leftrightarrow x + z \leq y + z$,

4. $x < y \Leftrightarrow x + z < y + z$,

5. $0 < x + y \Rightarrow 0 < x \vee 0 < y$,

6. $x < y \leq z \Rightarrow x < z$,

7. $x \leq y < z \Rightarrow x < z$,

8. $x \leq y \wedge 0 \leq z \Rightarrow xz \leq yz$,

9. $0 < z \Rightarrow (x < y \Leftrightarrow xz < yz)$,

10. $0 < 1$.

Proof. First, assume (†) and ($*$).

Items 1 and 2 are true by the convention after Lemma 4.1.7 that $\leq$ and # are defined in terms of $<$.

For item 4, suppose $x < y$, so $(x+z)-z < (y+z)-z$. By (†), we have $(x+z < y+z) \vee (-z < -z)$. The latter case contradicts irreflexivity of $<$.

Conversely, suppose $x+z < y+z$. By (†), we have $(x < y) \vee (z < z)$. Again, the latter case contradicts irreflexivity of $<$.

Item 3 follows from item 4 using the fact that if $A \Leftrightarrow B$ then $\neg A \Leftrightarrow \neg B$.

Item 5 is an instance of (†).

For item 6, suppose $x < y$ and $\neg(z < y)$. By cotransitivity, we have $x < z \vee z < y$, the latter case being excluded by assumption.

Item 7 goes through in a similar fashion.

For item 10, since 1 has multiplicative inverse 1, it is apart from 0, hence $0 < 1 \vee 1 < 0$. If $1 < 0$ then by item 4 we have $0 < -1$ and so by ($*$) we get $0 < (-1) \cdot (-1)$, that is, $0 < 1$, so by transitivity $1 < 1$, contradicting irreflexivity of $<$.

For item 8, suppose $x \leq y$ and $0 \leq z$ and $yz < xz$. Then $0 < z(x-y)$ by (†), and so, being apart from 0, $z(x-y)$ has a multiplicative inverse w. Hence z itself has a multiplicative inverse $w(x-y)$, and so $0 < z \vee z < 0$, where the latter case contradicts the assumption $0 \leq z$, so that we have $0 < z$. Now $w(x-y)$ has multiplicative inverse z, so it is apart from 0, that is $(0 < w(x-y)) \vee (w(x-y) < 0)$. In the latter case, from ($*$) we get $zw(x-y) < 0$, i.e. $1 < 0$ which contradicts item 10, so that we have $0 < w(x-y)$. By ($*$), from $0 < w(x-y)$ and $yz < xz$ we get $yzw(x-y) < xzw(x-y)$, so $y < x$, contradicting our assumption that $x \leq y$.

For item 9, assuming $0 < z$ and $x < y$, the required conclusion $xz < yz$ is an instance of ($*$).

For the other direction of item 9, assume $0 < z$ and $xz < yz$, so that $yz - xz$ has a multiplicative inverse w, and so z itself has multiplicative inverse $w(y-x)$. Then since $0 < z$ and $xz < yz$, by ($*$), we get $xzw(y-x) < yzw(y-x)$, and hence $x < y$.

Conversely, assume the 10 listed items—in particular, items 4, 5 and 9. In order to show (†), suppose $x + y < z + w$. So, by item 4, we get $(x+y) - (x+y) < (z+w) - (x+y)$, that is,

$0 < (z - x) + (w - y)$. By item 5, $(0 < z - x) \vee (0 < w - y)$, and so by item 4 in either case, we get $x < z \vee y < w$.

$(*)$ follows from item 9. □

Lemma 4.1.12. *An ordered field* $(F, 0, 1, +, \cdot, \min, \max, <)$ *is a constructive field* $(F, 0, 1, +, \cdot, \#)$.

Proof. We need to show that + is #-extensional, and that # is tight.

First, assume $w + x \# y + z$. We need to show $w \# y \vee x \# z$. Consider the case $w + x < y + z$, so that we can use (†) to obtain $w < y \vee x < z$, which gives $w \# y \vee x \# z$ in either case. The case $w + x > y + z$ is similar.

Tightness follows from the fact that $\leq$ is antisymmetric, combined with the fact that $\neg(P \vee Q)$ is equivalent to $\neg P \wedge \neg Q$. □

We will mainly be concerned with ordered fields, as opposed to the more general constructive fields. This is because the Archimedean property can be phrased straightforwardly for ordered fields, as in Section 4.3, and because the ordering relation allows us to define locators, as in Chapter 6.

We have defined ordered fields, which capture the algebraic structure of the real numbers.

4.2 Rationals

In Section 2.3 we introduced the natural numbers $\mathbb{N}$. We can define the integers as, for example,

$$\mathbb{Z} :\equiv \mathbb{N} + \mathbb{N}$$

where elements of the right summand represent nonnegative numbers, and elements of the left summand represent negative numbers.

Rational numbers can be presented as pairs $(k, n) : \mathbb{Z} \times \mathbb{N}_{\geq 1}$ of a numerator k and a positive denominator n. Such pairs can be reduced to their lowest terms with an idempotent map

$$\text{fracred} : \mathbb{Z} \times \mathbb{N}_{\geq 1} \to \mathbb{Z} \times \mathbb{N}_{\geq 1}.$$

The rationals $\mathbb{Q}$ can then be defined as the type $(\Sigma p : \mathbb{Z} \times \mathbb{N}_{\geq 1})\, \mathrm{fracred}(p) = p$ of fixpoints of this map. We also have the type $\mathbb{Q}_+$ of positive rationals.

The rationals form an ordered field. Its apartness relation is then equivalent to the negation of equality.

4.3 Archimedean property

We now define the notion of Archimedean ordered fields. We phrase this in terms of a certain interpolation property, that can be defined from the fact that there is a unique morphism of ordered fields from the rationals to every ordered field.

Definition 4.3.1. A morphism from an ordered field $(F, 0_F, 1_F, +_F, \cdot_F, \min_F, \max_F, <_F)$ to an ordered field $(G, 0_G, 1_G, +_G, \cdot_G, \min_G, \max_G, <_G)$ is a map $f : F \to G$ such that

1. f is a morphism of rings,

2. f reflects $<$ in the sense that for every $x, y : F$
$$f(x) <_G f(y) \Rightarrow x <_F y.$$

> *Remark* 4.3.2. The contrapositive of reflecting $<$ means preserving $\leq$.

Lemma 4.3.3. *For every ordered field $(F, 0_F, 1_F, +_F, \cdot_F, \min_F, \max_F, <_F)$, there is a unique morphism i of ordered fields from the rationals to F. Additionally, i* preserves $<$ *in the sense that for every $q, r : \mathbb{Q}$*
$$q < r \Rightarrow i(q) <_F i(r).$$

Proof. i can be constructed in the usual way by first considering what the integers get mapped to, and then considering arbitrary rationals.

We can also show that i is unique in the usual way.

To see that i preserves $<$, let k/n and k'/n' be two rationals, with $k, k' : \mathbb{Z}$ and $n, n' : \mathbb{N}_+$, and assume $k/n < k'/n'$. Hence $kn' < k'n$ and so $k'n = kn' + m$ for some $m : \mathbb{N}_+$. By induction on m, it follows that $i(kn') < i(k'n)$ and hence $i(k/n) < i(k'/n')$ using the fact that i is a morphism of rings.

Since i preserves $<$, it also reflects it: consider rationals $q, r : \mathbb{Q}$, and suppose $i(q) <_F i(r)$. By trichotomy of the rationals, we have $(q < r) \vee (q = r) \vee (r < q)$. The first disjunct is the desired conclusion, and the latter two contradict that $<_F$ is a strict order since they yield, respectively, $i(q) <_F i(q)$ and $i(r) <_F i(q)$. □

> *Remark* 4.3.4. Since i preserves $<$, it is an embedding.

Definition 4.3.5. Let $(F, 0_F, 1_F, +_F, \cdot_F, \min_F, \max_F, <_F)$ be an ordered field, so that we get a canonical morphism $i : \mathbb{Q} \to F$ of ordered fields, as in Lemma 4.3.3. We say the ordered field $(F, 0_F, 1_F, +_F, \cdot_F, \min_F, \max_F, <_F)$ is *Archimedean* if

$$(\forall x, y : F)(\exists q : \mathbb{Q}) x < i(q) < y.$$

If the ordered field is clear from the context, we will identify rationals $q : \mathbb{Q}$ with their inclusion $i(q)$ in the ordered field, so that we may also say that $(F, 0_F, 1_F, +_F, \cdot_F, \min_F, \max_F, <_F)$ is Archimedean if

$$(\forall x, y : F)(\exists q : \mathbb{Q}) x < q < y.$$

Example 4.3.6. In an Archimedean ordered field, all numbers are bounded by rationals. That is, for a given $x : F$, there exist $q, r : \mathbb{Q}$ with $q < x < r$. This follows from applying the Archimedean property to $x - 1 < x$ and $x < x + 1$.

4.4 Cauchy completeness of real numbers

We focus on Cauchy completeness, rather than Dedekind or Dedekind-MacNeille completeness, as we will focus on the computation of digit expansions, for which Cauchy completeness suffices.

In order to state that an ordered field is Cauchy complete, we need to define when sequences are Cauchy, and when a sequence has a limit. We also take the opportunity to define the set of *Cauchy reals* in Definition 4.4.9. Surprisingly, this ordered field cannot be shown to be Cauchy complete.

Fix an ordered field $(F, 0_F, 1_F, +_F, \cdot_F, \min_F, \max_F, <_F)$. We get a notion of distance, given by the absolute value as

$$|x - y| \coloneqq \max_F(x - y, -(x - y)).$$

Consider a sequence $x : \mathbb{N} \to F$ of elements of F. Classically, we may state that x is Cauchy as

$$(\forall \varepsilon : \mathbb{Q}_+)(\exists N : \mathbb{N})(\forall m, n : \mathbb{N}) m, n \geq N \Rightarrow |x_m - x_n| < \varepsilon,$$

We can interpret the quantifiers as in Definition 2.4.5.

Following a propositions-as-types interpretation, we may also state that x is Cauchy as the structure

$$(\Pi \varepsilon : \mathbb{Q}_+)(\Sigma N : \mathbb{N})(\Pi m, n : \mathbb{N}) m, n \geq N \to |x_m - x_n| < \varepsilon.$$

The dependent sum represents a choice of index N for every error ε, and so we have arrived at the following definition.

Definition 4.4.1. For a sequence of reals $x : \mathbb{N} \to F$, a *a modulus of Cauchy convergence* is a map $M : \mathbb{Q}_+ \to \mathbb{N}$ such that

$$(\forall \varepsilon : \mathbb{Q}_+)(\forall m, n : \mathbb{N}) m, n \geq M(\varepsilon) \Rightarrow |x_m - x_n| < \varepsilon.$$

In constructive mathematics, we typically use such sequences with modulus, for example, because they can sometimes be used to compute limits of Cauchy sequences, avoiding choice axioms.

Definition 4.4.2. A number $l : F$ is the *limit* of a sequence $x : \mathbb{N} \to F$ if the sequence converges to l in the usual sense:

$$(\forall \varepsilon : \mathbb{Q}_+)(\exists N : \mathbb{N})(\forall n : \mathbb{N}) n \geq N \Rightarrow |x_n - l| < \varepsilon.$$

Remark 4.4.3. We do not consider the statement of convergence in propositions-as-types

$$(\Pi\varepsilon : \mathbb{Q}_+)(\Sigma N : \mathbb{N})(\Pi n : \mathbb{N}) n \geq N \to |x_n - l| < \varepsilon,$$

because if the sequence has a modulus of Cauchy convergence M, then $\lambda\varepsilon.M(\varepsilon/2)$ is a modulus of convergence to the limit l, so that we get an element of the above type.

Definition 4.4.4. The ordered field $(F, 0_F, 1_F, +_F, \cdot_F, \min_F, \max_F, <_F)$ is said to be *Cauchy complete* if for every sequence x with modulus of Cauchy convergence M, we have a limit of x.

In other words, an ordered field is Cauchy complete iff from a sequence–modulus pair (x, M), we can compute a limit of x.

For the remainder of this section, additionally assume that F is Archimedean.

Lemma 4.4.5. *The type of limits of a fixed sequence $x : \mathbb{N} \to F$ is a proposition.*

Proof. This can be shown using the usual proof that limits are unique in Archimedean ordered fields, followed by an application of Lemma 2.6.20. □

Corollary 4.4.6. *Fix a given sequence $x : \mathbb{N} \to F$. Suppose that we know that there* exists *a limit of the sequence. Then we can* compute *a limit of the sequence.*

Proof. By applying the induction principle of propositional truncations of Definition 2.4.3. □

Corollary 4.4.7. *Fix a given sequence $x : \mathbb{N} \to F$. Suppose that, from a modulus of Cauchy convergence, we can compute a limit of the sequence. Then from the* existence *of the modulus of Cauchy convergence we can compute a limit of the sequence.*

Proof. By applying the induction principle of propositional truncations of Definition 2.4.3. □

We can thus compute the limit of $x : \mathbb{N} \to F$ as the number $\lim(x, p)$, where p is a proof that the limit of x exists. We will rather use the more traditional notation $\lim_{n\to\infty} x_n$ for this number.

Example 4.4.8 (Exponential function)**.** In a Cauchy complete Archimedean ordered field, we can define an exponential function $\exp : F \to F$ by $\exp(x) = \sum_{k=0}^{\infty} \frac{x^k}{k!}$. For a given input x, we obtain the *existence* of a modulus of Cauchy convergence for the output from boundedness of x, that is, from the fact that $(\exists q, r : \mathbb{Q}) q < x < r$.

The point of this work is that, because we have a single language for properties and structure, we can see more precisely what is needed for certain computations. In the above example, we explicitly do not require that inputs come *equipped* with a modulus of Cauchy convergence, but rather that there *exists* such a modulus. On the one hand, we do need a modulus to obtain the limit, but as the limit *value* is independent of the chosen modulus, existence of such a modulus suffices.

Definition 4.4.9. The *Cauchy reals* $\mathbb{R}_\mathrm{C}$ is the collection of rational sequences equipped with a modulus of Cauchy convergence, quotiented (as in Section 2.7) by an equivalence relation that relates two sequence–modulus pairs (x, M) and (y, N) iff

$$(\forall \varepsilon : \mathbb{Q}_+) \left| x_{M(\varepsilon/4)} - y_{M(\varepsilon/4)} \right| < \varepsilon.$$

The Cauchy reals form an Archimedean ordered field in a natural way. The natural strategy to prove that the Cauchy reals are Cauchy complete, perhaps surprisingly, does not work, and in some constructive foundations the Cauchy completeness of the Cauchy reals is known to be false [68].

> One immediate source of confusion here is identifying reals with sequence–modulus pairs. A real is an equivalence class of such pairs, and it is not obvious how a representative can be chosen constructively from each real; in fact, this cannot in general be done, as we shall see. This distinction has not always be[en] made though. For instance, as observed by Fred Richman, in [92], the Cauchy completeness of the reals was stated as a theorem, but what was proved was the Cauchy completeness of sequence–modulus pairs. To be precise, what was shown was that, given a countable sequence, with its own modulus of convergence, of sequence–modulus pairs, then there is a limit sequence, with modulus. For that matter, it is not hard (and left to the reader) that, even if the given sequence does not come equipped with its own

> modulus, it still has a Cauchy sequence as a limit, although we will have to punt on the limit having a modulus. But neither of those two observations is the Cauchy completeness of the reals.
>
> — *Lubarsky in "On the Cauchy Completeness of the Constructive Cauchy Reals" [68]*

An alternative interpretation of the non-completeness of the Cauchy reals is that the sequential definition of completeness ought to be amended [80].

4.5 HoTT book reals

We will now define a Cauchy complete type of Cauchy sequences. However, as opposed to taking Cauchy sequences valued in the rationals, we will define the *HoTT book reals* $\mathbb{R}_{\mathbf{H}}$, with Cauchy sequences valued in $\mathbb{R}_{\mathbf{H}}$ itself, so that we avoid needing to pick representatives.

> *Remark* 4.5.1. The name "HoTT book reals" refers to a common way to refer to The Univalent Foundations Program [91], in which the name *Cauchy reals* is used. We rather use this to refer to the quotient type of Section 4.4.

Following Sojakova [88], we take an algebraic view on types of real numbers and higher inductive-inductive types (HIITs). We will do this, as opposed to directly giving the type-theoretic inference rules of the HIIT, in order to make a clear link with the Euclidean reals in Section 5.1. By analogy with Richman [80] and the The Univalent Foundations Program [91], we define premetric spaces and ε-closeness.

Definition 4.5.2. A *premetric* on a type $R : \mathcal{U}$ is a relation

$$\cdot \sim_{\cdot} \cdot : \mathbb{Q}_{+} \times R \times R \to \mathrm{HProp}.$$

We will often write R for the *premetric space* $(R, \sim)$, leaving the premetric $\sim$ implicit. In the case that $u \sim_{\varepsilon} v$ holds (where $\varepsilon : \mathbb{Q}_{+}$ and $u, v : R$) we say that u and v are *ε-close.*

Note the outright lack of natural conditions one might put on $\sim$: our premetric spaces are a very wild notion. In fact, having few conditions here is a good thing, as any conditions introduced now would need to be respected later by the induction principle in Definition 4.5.12, thus making that induction principle harder to use.

We think of $u \sim_\varepsilon v$ as being true when $|u - v| < \varepsilon$. Once operations have been defined for the HoTT book reals, then this can be made precise [91, Theorem 11.3.44]. In Chapter 8 we work with a premetric defined in terms of a (pseudo)metric as $u \sim_\varepsilon v :\equiv \rho(u, v) < \varepsilon$.

Definition 4.5.3. For a premetric space $(X, \sim)$, and a sequence $x : \mathbb{N} \to X$ in X, a *modulus of Cauchy convergence* for $x : \mathbb{N} \to X$ is an assignment $M : \mathbb{Q}_+ \to \mathbb{N}$ such that

$$(\forall \varepsilon : \mathbb{Q}_+)(\forall m, n : \mathbb{N}) m, n \geq M(\varepsilon) \Rightarrow x_m \sim_\varepsilon x_n.$$

When the premetric is defined in terms of the absolute value as $|x - y| < \varepsilon$, this coincides with the modulus of Cauchy convergence in Definition 4.4.1. As in Section 4.4, we are mainly interested in sequences that are Cauchy with modulus, where the modulus is either given explicitly, or proved to exist in the sense of Definition 2.4.5, as opposed to sequences that are Cauchy convergent in the classical sense.

Definition 4.5.4. For a premetric space $(X, \sim)$, and a sequence $x : \mathbb{N} \to X$ in X, a point $x_\infty : X$ is a *limit* of x if

$$(\forall \varepsilon : \mathbb{Q}_+)(\exists N : \mathbb{N})(\forall n : \mathbb{N}) n \geq N \Rightarrow x_n \sim_\varepsilon x_\infty.$$

Again, this definition is justified because it coincides with Definition 4.4.2.

So far we have taken a sequence in R to be a function $\mathbb{N} \to R$, and a modulus of Cauchy convergence to be a function $\mathbb{Q}_+ \to \mathbb{N}$. The following notion encapsulates the composition $\mathbb{Q}_+ \to R$ of a modulus of convergence with a sequence.

Definition 4.5.5. If R is a premetric space, then $x : \mathbb{Q}_+ \to R$ is a *Cauchy approximation* if

$$\text{isCauchy}(x) :\equiv (\forall \delta, \varepsilon : \mathbb{Q}_+) x_\delta \sim_{\delta+\varepsilon} x_\varepsilon. \tag{4.1}$$

We define the type C_R of Cauchy approximations in R as

$$C_R :\equiv (\Sigma x : \mathbb{Q}_+ \to R) \, \text{isCauchy}(x).$$

Since being a Cauchy approximation is a property rather than structure, we identify elements of C_R with their underlying map $\mathbb{Q}_+ \to R$.

Remark 4.5.6. Compared to Definition 4.5.3, for Cauchy approximations, the Cauchy-ness is specified by *proposition* (4.1), rather than the *structure* of a modulus, just like being a *regular* Cauchy sequence can be phrased as a proposition. In exchange, when constructing the underlying data $\mathbb{Q}_+ \to R$, we need to be aware of the convergence rate, whereas with sequences we can construct the underlying data $\mathbb{N} \to R$ and consider the rate of convergence only later when constructing a modulus for it. As in Example 4.4.8, the exponential function on real numbers is more easily defined as a power series, since the rate of convergence of a power series depends on the magnitude of the input value.

Definition 4.5.7. If x is a Cauchy approximation in a premetric space R, then we say that $u : R$ is a *limit* of x if

$$(\forall \varepsilon, \theta : \mathbb{Q}_+) x_\varepsilon \sim_{\varepsilon+\theta} u.$$

If for every Cauchy approximation we can compute a limit, we say that R is *Cauchy complete.*

Cauchy approximations and sequences with modulus of Cauchy convergence (as in Definition 4.4.1) are interdefinable in the sense that from one we can compute the other, such that, in particular, they have the same limit, if any. We give a construction in the proof of Corollary 4.5.11.

In our very weak notion of premetric spaces, we do not automatically have uniqueness of limits, so that the *existence* of limits does not imply that we can *compute* limits.

Definition 4.5.8. A *Cauchy structure* is a premetric space $(R, \sim)$ together with the following structure, collected in a Σ-type.

$$\begin{aligned}
\mathrm{rat} &: \mathbb{Q} \to R \\
\lim &: C_R \to R \\
\mathrm{eq} &: (\Pi u, v : R)\,((\forall \varepsilon : \mathbb{Q}_+) u \sim_\varepsilon v) \to u =_R v \\
&(\Pi q, r : \mathbb{Q})(\Pi \varepsilon : \mathbb{Q}_+)(-\varepsilon < q - r < \varepsilon) \to \mathrm{rat}(q) \sim_\varepsilon \mathrm{rat}(r) \\
&(\Pi q : \mathbb{Q})(\Pi y : C_R)(\Pi \varepsilon, \delta : \mathbb{Q}_+)\,\mathrm{rat}(q) \sim_\varepsilon y_\delta \to \mathrm{rat}(q) \sim_{\varepsilon+\delta} \lim(y) \\
&(\Pi x : C_R)(\Pi r : \mathbb{Q})(\Pi \varepsilon, \delta : \mathbb{Q}_+) x_\delta \sim_\varepsilon \mathrm{rat}(r) \to \lim(x) \sim_{\varepsilon+\delta} \mathrm{rat}(r) \\
&(\Pi x : C_R)(\Pi y : C_R)(\Pi \varepsilon, \delta, \eta : \mathbb{Q}_+) x_\delta \sim_\varepsilon y_\eta \to \lim(x) \sim_{\varepsilon+\delta+\eta} \lim(y)
\end{aligned}$$

A morphism of Cauchy structures from R to S is a map $f : R \to S$ and a family of maps $g_{\varepsilon,u,v} : u \sim_\varepsilon v \to f(u) \sim_\varepsilon f(v)$ that preserve rat, lim and eq in the obvious sense. Explicitly:

$$\begin{aligned}
\text{CS-hom}(R, S) \coloneqq &(\Sigma f : R \to S) \\
&\quad (\Sigma g : (\Pi \varepsilon : \mathbb{Q}_+)(\Pi u, v : R) u \sim_\varepsilon v \;\to\; f(u) \sim_\varepsilon f(v)) \\
&\quad ((\Pi q : \mathbb{Q}) f(\mathrm{rat}(q)) = \mathrm{rat}(q)) \\
&\quad \times ((\Pi x : C_R) f(\lim(x)) = \lim(f \circ x)) \\
&\quad \times ((\Pi u, v : R)(\Pi p : (\forall \varepsilon : \mathbb{Q}_+) u \sim_\varepsilon v) f^*(\mathrm{eq}(u, v, p)) = \mathrm{eq}(f(u), f(v), g(p)))\,.
\end{aligned}$$

Remark 4.5.9.

1. The remaining four maps of the Cauchy structure are automatically preserved as $\sim$ is valued in propositions.

2. A morphism of Cauchy structures from R to S gives rise to a map $C_R \to C_S$.

3. Identity maps are Cauchy structure morphisms, and Cauchy structure morphisms are closed under composition.

4. We emphasize that even though a Cauchy structure has the lim map, it need not be Cauchy complete, since the elements x_ε of a Cauchy approximation might not be of the form $\mathrm{rat}(q)$ or $\lim(z)$. In other words, the lim map does not necessarily compute limits.

   For example, we may define a Cauchy structure on a type $\mathbf{2}$ with two elements, where both rat and lim constantly output $\mathtt{ff}$, and we have the relations $\mathtt{tt} \sim_\varepsilon \mathtt{tt}$ and $\mathtt{ff} \sim_\varepsilon \mathtt{ff}$ for all ε, but nothing else. Then we have a Cauchy approximation that is constantly $\mathtt{tt}$, and lim computes it limit as $\mathtt{ff}$, which is not a limit in the sense of Definition 4.5.7—a valid limit would be $\mathtt{tt}$.

A Cauchy complete Archimedean ordered field induces a canonical Cauchy structure. We use the following lemma.

Lemma 4.5.10. *Let* $(F, 0, 1, +, \cdot, \min, \max, <)$ *be an Archimedean ordered field, and define a premetric on* F *by* $u \sim_\varepsilon v :\equiv |u - v| < \varepsilon$. *For a Cauchy approximation* $x : C_F$ *in* F, *a limit* $u : F$ *of* x *in the sense of Definition 4.5.7, and* $\delta : \mathbb{Q}_+$, *we have* $|x_\delta - u| \leq \delta$.

Proof. In order to show $|x_\delta - u| \leq \delta$, we aim to show a contradiction from the assumption $\delta < |x_\delta - u|$. By the Archimedean property there exists $\eta : \mathbb{Q}_+$ with

$$\delta < \delta + \eta < |x_\delta - u| \,.$$

Since u is a limit of x_δ, we have $|x_\delta - u| < \delta + \eta$, which contradicts the above. □

Corollary 4.5.11. *A Cauchy complete Archimedean ordered field* $(F, 0, 1, +, \cdot, \min, \max, <)$ *with the premetric* $u \sim_\varepsilon v :\equiv |u - v| < \varepsilon$ *comes equipped with a canonical Cauchy structure.*

Proof. The map $\mathrm{rat} : \mathbb{Q} \to F$ is obtained from Lemma 4.3.3.

To construct the lim map, take a Cauchy approximation $x : \mathbb{Q}_+ \to F$. We can present this as a sequence–modulus pair by setting $y_n :\equiv x_{1/n}$ with $M(\varepsilon) :\equiv \left\lceil \frac{2}{\varepsilon} \right\rceil$. Now (y, M) has a limit l by Cauchy completeness in the sense of Definition 4.4.4, which is then also a limit of x in the sense of Definition 4.5.7. Incidentally, we can conversely present a sequence y with Cauchy modulus M as a Cauchy approximation, namely by setting $x_\varepsilon :\equiv y_{M(\varepsilon)}$, although we will not use this here.

We construct $\mathrm{eq} : (\Pi u, v : F)\,((\forall \varepsilon : \mathbb{Q}_{+}) u \sim_{\varepsilon} v) \rightarrow u = v$. Suppose that u and v are ε-close for any $\varepsilon : \mathbb{Q}_{+}$. We show $u = v$ using antisymmetry of $\leq$, so that it suffices to show $u \leq v$ and $v \leq u$. Without loss of generality, we show the former, for which it suffices to show a contradiction from $v < u$. By using the Archimedean property on $0 < u - v$, there exists $\varepsilon : \mathbb{Q}$ with $0 < \varepsilon < u - v$, contradicting that u and v are ε-close.

The first distance law follows from the fact that the map rat obtained from Lemma 4.3.3 is a morphism of ordered fields, so that $\mathrm{rat}(q-r) = \mathrm{rat}(q) - \mathrm{rat}(r)$. The remaining three distance laws can be shown by applying Lemma 4.5.10: for instance: if $|\mathrm{rat}(q) - y_{\delta}| < \varepsilon$, then since $|y_{\delta} - \lim y| \leq \delta$, we get $|\mathrm{rat}(q) - \lim y| < \varepsilon + \delta$, showing the second distance law. □

We now define the HoTT book reals $\mathbb{R}_{\mathrm{H}}$ [91, Section 11.3]. We use the following definition, which is equivalent to the one in The Univalent Foundations Program [91] as we will show in Theorem 4.5.18.

Definition 4.5.12. $\mathbb{R}_{\mathrm{H}}$ is a homotopy-initial Cauchy structure, in the sense that for any other Cauchy structure S, the type of Cauchy structure morphisms from $\mathbb{R}_{\mathrm{H}}$ to S is contractible.

Theorem 4.5.13. $\mathbb{R}_{\mathrm{H}}$ *is Cauchy complete.*

To prove this theorem, we will develop an induction principle for $\mathbb{R}_{\mathrm{H}}$, so that $\mathbb{R}_{\mathrm{H}}$ is equivalent, and hence by univalence, identical, to the type developed in [91, Section 11.3].

Definition 4.5.14. Given

$$
\begin{aligned}
&A : \mathbb{R}_{\mathrm{H}} \rightarrow \mathcal{U} \\
&B : (\Pi u, v : \mathbb{R}_{\mathrm{H}}) A(u) \rightarrow A(v) \rightarrow (\Pi \varepsilon : \mathbb{Q}_{+})(u \sim_{\varepsilon} v) \rightarrow \mathrm{HProp}
\end{aligned}
$$

we obtain a natural premetric on $(\Sigma u : \mathbb{R}_{\mathrm{H}}) A(u)$, given by the relation:

$$(u, a) \sim_{\varepsilon} (v, b) :\equiv (\Sigma \zeta : u \sim_{\varepsilon} v) B(u, v, a, b, \varepsilon, \zeta)$$

For the remainder of this section, fix a choice of $A : \mathbb{R}_{\mathrm{H}} \rightarrow \mathcal{U}$ and $B : (\Pi u, v : \mathbb{R}_{\mathrm{H}}) A(u) \rightarrow A(v) \rightarrow (\Pi \varepsilon : \mathbb{Q}_{+})(u \sim_{\varepsilon} v) \rightarrow \mathrm{HProp}$ — these type families will be input for our induction principle. The remaining input will allow us to define a Cauchy structure on $(\Sigma u : \mathbb{R}_{\mathrm{H}}) A(u)$.

We will often denote the type $B(u, v, a, b, \varepsilon, \zeta)$ by $a \sim_\varepsilon b$, since u can typically be inferred from a and v from b, and ζ is unique since the premetric on $\mathbb{R}_\mathrm{H}$ is valued in propositions.

Definition 4.5.15. Let $x : C_{\mathbb{R}_\mathrm{H}}$ and $a : (\Pi\varepsilon : \mathbb{Q}_+)A(x_\varepsilon)$, satisfying

$$(\forall \delta, \varepsilon : \mathbb{Q}_+) a_\delta \sim_{\delta+\varepsilon} a_\varepsilon.$$

Then we call a a *dependent Cauchy approximation* over x. We denote the type of all dependent Cauchy approximations over x by $\mathcal{D}^x_A$, and again identify its elements with their underlying (dependent) function.

Lemma 4.5.16. *Suppose $x : C_{\mathbb{R}_\mathrm{H}}$ and $a : (\Pi\varepsilon : \mathbb{Q}_+)A(x_\varepsilon)$. Then the function*

$$\lambda\varepsilon.(x_\varepsilon, a_\varepsilon)$$

is a Cauchy approximation in $(\Sigma u : \mathbb{R}_\mathrm{H})A(u)$ iff a is a dependent Cauchy approximation over x.

Proof. Straightforward. □

The above lemma allows us to take limits componentwise, as we will do in the proof of an induction principle in Theorem 4.5.18. To be able to phrase an induction principle, we first define dependent identifications, namely the identity of elements in a type family evaluated at identical elements of $\mathbb{R}_\mathrm{H}$.

Definition 4.5.17. Given a type $A : \mathcal{U}$, a type family $B : A \to \mathcal{U}$, an identification $p : x =_A y$ in A, and elements $u : B(x)$ and $v : B(y)$, the type of *dependent identifications $u =^B_p v$* is defined by induction on p: if p is $\mathsf{refl}(x)$ then $(u =^B_{\mathsf{refl}(x)} v) :\equiv (u =_{B(x)} v)$. We refer to elements of $u =^B_p v$ as *identifications from u to v over p*.

In particular, an identification $p : x =_A y$ can be combined with a dependent identification $q : u =^B_p v$ into an identification $(x, u) =_{(\Sigma a:A)B(a)} (y, v)$ in the dependent sum type, and vice versa an identification in the dependent sum type gives rise to an identification p in A and a dependent identification over p.

Theorem 4.5.18. *Suppose we are provided*

$$\begin{aligned}
&A : \mathbb{R}_{\mathrm{H}} \to \mathcal{U} \\
&B : (\Pi u, v : \mathbb{R}_{\mathrm{H}}) A(u) \to A(v) \to (\Pi \varepsilon : \mathbb{Q}_{+})(u \sim_{\varepsilon} v) \to \mathrm{HProp}
\end{aligned}$$

and the following data.

$$\begin{aligned}
&f_{\mathrm{rat}} : (\Pi q : \mathbb{Q}) A(\mathrm{rat}(q)) \\
&f_{\mathrm{lim}} : (\Pi x : C_{\mathbb{R}_{\mathrm{H}}}) \mathcal{D}^{x}_{A} \to A(\lim(x)) \\
&f_{\mathrm{eq}} : (\Pi u, v : \mathbb{R}_{\mathrm{H}})(\Pi a : A(u))(\Pi b : A(v))(\Pi p : (\forall \varepsilon : \mathbb{Q}_{+}) a \sim_{\varepsilon} b) a =^{A}_{\mathrm{eq}(u,v,p)} b \\
&\quad (\Pi q, r : \mathbb{Q})(\Pi \varepsilon : \mathbb{Q}_{+}) - \varepsilon < q - r < \varepsilon \to f_{\mathrm{rat}}(q) \sim_{\varepsilon} f_{\mathrm{rat}}(r) \\
&\quad (\Pi q : \mathbb{Q})(\Pi y : C_{\mathbb{R}_{\mathrm{H}}})(\Pi b : \mathcal{D}^{y}_{A})(\Pi \delta, \varepsilon : \mathbb{Q}_{+}) \,\mathrm{rat}(q) \sim_{\varepsilon} y_{\delta} \\
&\qquad\quad \to f_{\mathrm{rat}}(q) \sim_{\varepsilon} b_{\delta} \to f_{\mathrm{rat}}(q) \sim_{\varepsilon+\delta} f_{\mathrm{lim}}(y, b) \\
&\quad (\Pi x : C_{\mathbb{R}_{\mathrm{H}}})(\Pi a : \mathcal{D}^{x}_{A})(\Pi r : \mathbb{Q})(\Pi \delta, \varepsilon : \mathbb{Q}_{+}) x_{\delta} \sim_{\varepsilon} \mathrm{rat}(r) \\
&\qquad\quad \to a_{\delta} \sim_{\varepsilon} f_{\mathrm{rat}}(r) \to f_{\mathrm{lim}}(x, a) \sim_{\varepsilon+\delta} f_{\mathrm{rat}}(r) \\
&\quad (\Pi x, y : C_{\mathbb{R}_{\mathrm{H}}})(\Pi a : \mathcal{D}^{x}_{A})(\Pi b : \mathcal{D}^{y}_{A})(\Pi \delta, \eta, \varepsilon : \mathbb{Q}_{+}) x_{\delta} \sim_{\varepsilon} y_{\eta} \\
&\qquad\quad \to a_{\delta} \sim_{\varepsilon} b_{\eta} \to f_{\mathrm{lim}}(x, a) \sim_{\varepsilon+\delta+\eta} f_{\mathrm{lim}}(y, b)
\end{aligned}$$

In that case, we obtain

$$\begin{aligned}
&f : (\Pi u : \mathbb{R}_{\mathrm{H}}) A(u) \qquad \textit{and} \\
&g : (\Pi u, v : \mathbb{R}_{\mathrm{H}})(\Pi \varepsilon : \mathbb{Q}_{+})(\Pi \zeta : x \sim_{\varepsilon} y) B(u, v, f(u), f(v), \varepsilon, \zeta),
\end{aligned}$$

satisfying

$$\begin{aligned}
&f(\mathrm{rat}(q)) = f_{\mathrm{rat}}(q) \qquad \textit{and} \\
&f(\lim(x)) = f_{\mathrm{lim}}(x, (f, g)[x]),
\end{aligned}$$

where $(f, g)[x]$ *is the dependent Cauchy approximation defined by*

$$(f, g)[x]_{\varepsilon} :\equiv f(x_{\varepsilon}).$$

Proof. We reason similar to Sojakova [88]. Write $T = (\Sigma u : \mathbb{R}_{\mathbf{H}})A(u)$. Given the input data, we can define a natural Cauchy structure on T. For example, $\mathrm{rat}_T(q) :\equiv (\mathrm{rat}(q), f_{\mathrm{rat}}(q))$.

Hence, by homotopy-initiality of $\mathbb{R}_{\mathbf{H}}$, we obtain $h : \mathbb{R}_{\mathbf{H}} \to T$ and $i_{\varepsilon,u,v} : u \sim_\varepsilon v \to h(u) \sim_\varepsilon h(v)$ preserving rat, lim and eq in the obvious sense.

Postcomposing h and i (the latter componentwise) with the first projection functions gives us a Cauchy morphism $\mathbb{R}_{\mathbf{H}} \to \mathbb{R}_{\mathbf{H}}$, and so by homotopy-initiality, the first component of any $h(u)$ is identical to u. By transporting along this identification, we obtain dependent functions f and g with the required properties. □

We have shown that $\mathbb{R}_{\mathbf{H}}$ satisfies the same universal property as the type defined in [91, Section 11.3], so that the types are equivalent. We now appeal to The Univalent Foundations Program [91, Section 11.3.2] for a proof of Theorem 4.5.13.

4.6 Dedekind reals

The set of Dedekind cuts is another Cauchy complete Archimedean ordered field. Recall the notation $x \in B$ for a subtype $B : \mathcal{P}X$ of a type $X\mathcal{U}$ from Section 2.6. A Dedekind real is defined by a pair (L, U) of predicates $\mathcal{P}\mathbb{Q}$ on $\mathbb{Q}$ with some properties. To phrase these properties succinctly, we use the following notation for $x = (L, U)$:

$$(q < x) :\equiv (q \in L) \qquad \text{and}$$
$$(x < r) :\equiv (r \in U).$$

This notation will be justified by the fact that $q \in L$ holds iff $i(q) < x$, with i the inclusion of the rationals into the Dedekind reals from Lemma 4.3.3.

There also exist one-sided definitions of Dedekind cuts [92, Chapter 5, Definition 5.1]. However, two-sided cuts will be more convenient for the definition of locators in Chapter 6.

Definition 4.6.1. A pair $x = (L, U)$ of predicates on the rationals is a *Dedekind cut* or *Dedekind real* if it satisfies the four Dedekind properties:

1. *bounded:* $(\exists q : \mathbb{Q})q < x$ and $(\exists r : \mathbb{Q})x < r$.

2. *rounded:* For all $q, r : \mathbb{Q}$,

$$q < x \Leftrightarrow (\exists q' : \mathbb{Q})(q < q') \wedge (q' < x) \qquad \text{and}$$
$$x < r \Leftrightarrow (\exists r' : \mathbb{Q})(r' < r) \wedge (x < r').$$

3. *transitive:* $(q < x) \wedge (x < r) \Rightarrow (q < r)$ for all $q, r : \mathbb{Q}$.

4. *located:* $(q < r) \Rightarrow (q < x) \vee (x < r)$ for all $q, r : \mathbb{Q}$.

The collection $\mathbb{R}_\mathrm{D}$ of pairs of predicates (L, U) together with proofs of the four properties, collected in a Σ-type, is called the *Dedekind reals.*

Remark 4.6.2. The Univalent Foundations Program [91] has *disjointness*

$$(\forall q : \mathbb{Q})\neg(x < q \wedge q < x)$$

instead of the transitivity property, which is equivalent to it in the presence of the other conditions, and it is this disjointness condition that we use most often in proofs.

Proof. Assuming transitivity, if $x < q \wedge q < x$, then transitivity yields $q < q$, which contradicts irreflexivity of $<$ on the rationals, which shows disjointness.

Conversely, if $q < x$ and $x < r$, apply trichotomy of the rationals on q and r: in case that $q < r$ we are done, and in the other two cases we obtain $x < q$, contradicting disjointness. □

Lemma 4.6.3. *The type $\mathbb{R}_\mathrm{D}$ is a set.*

Proof. By Lemma 2.6.19, it suffices to show that $(\mathbb{Q} \to \mathrm{HProp}) \times (\mathbb{Q} \to \mathrm{HProp})$ is a set. By the characterization of identity types in $\times$, this follows from the fact that $\mathbb{Q} \to \mathrm{HProp}$ is a set. This, in turn, using function extensionality, follows from the fact that HProp is a set, which holds by propositional univalence: namely, for $P, Q : \mathrm{HProp}$, the type $P = Q$ is equivalent to $P \simeq Q$, which is equivalent to $P \Leftrightarrow Q$, which is a proposition. □

Discussion 4.6.4. We quickly consider the universe levels. If the rationals are developed in $\mathcal{U}_i$, and we take subsets to mean maps to propositions in the same universe $\mathcal{U}_i$, then a pair of predicates on $\mathbb{Q}$ is an element of $\mathcal{P}_i\mathbb{Q} \times \mathcal{P}_i\mathbb{Q}$. The condition that (L, U) is a Dedekind real can be stated as a proposition in HProp_i, that is, we have a map $\mathrm{isCut} : \mathcal{P}_i\mathbb{Q} \times \mathcal{P}_i\mathbb{Q} \to \mathrm{HProp}_i$ expressing the proposition $\mathrm{isCut}(L, U)$ that (L, U) is a Dedekind cut. The type $(\Sigma L, U : \mathcal{P}_i\mathbb{Q})\,\mathrm{isCut}(L, U)$ of all Dedekind reals is then a type in $\mathcal{U}_{i+1}$. We discuss the definition of isCut in more detail in Section 6.10.

Definition 4.6.5. For Dedekind reals x and y, we define the strict ordering relation by

$$x < y :\equiv (\exists q : \mathbb{Q}) x < q < y$$

where $x < q < y$ means $(x < q) \land (q < y)$, and their *apartness* by

$$x \# y :\equiv (x < y) \lor (y < x).$$

As is typical in constructive analysis, we have $x \# y \Rightarrow \neg(x = y)$, but not the converse.

The Dedekind reals form an ordered field. We will not show this in detail, but for instance, addition can be defined by

$$\begin{aligned} (q < x + y) &:\equiv (\exists s, t : \mathbb{Q})(q = s + t) \land (s < x) \land (t < y) \quad \text{and} \\ (x + y < r) &:\equiv (\exists s, t : \mathbb{Q})(r = s + t) \land (x < s) \land (y < t), \end{aligned}$$

where it has to be checked that $x + y$ satisfies the conditions of Dedekind cuts as in Definition 4.6.1. Similarly, the constant $0_{\mathbb{R}_\mathrm{D}} : \mathbb{R}_\mathrm{D}$ representing zero can be defined neatly by $q < 0_{\mathbb{R}_\mathrm{D}} :\equiv q < 0$ and $0_{\mathbb{R}_\mathrm{D}} < r :\equiv 0 < r$, where the inequalities on the right-hand side refer to the ordering on $\mathbb{Q}$.

Because of our chosen strict ordering relation, the inequality $<$ of the Dedekind reals automatically satisfies the Archimedean property, using the fact stated above that $i(q) < x$ holds when q is in the lower cut of x, and similarly for $x < i(q)$. The following proof that $\mathbb{R}_\mathrm{D}$ is Cauchy complete is based on The Univalent Foundations Program [91, Theorem 11.2.12].

Lemma 4.6.6. *The Dedekind reals are Cauchy complete. More explicitly, given a modulus of Cauchy convergence for a sequence x of Dedekind reals, we can compute its limit l as the Dedekind cut defined by:*

$$(q < l) :\equiv (\exists \varepsilon, \theta : \mathbb{Q}_{+})(q + \varepsilon + \theta < x_{M(\varepsilon)}),$$
$$(l < r) :\equiv (\exists \varepsilon, \theta : \mathbb{Q}_{+})(x_{M(\varepsilon)} < r - \varepsilon - \theta).$$

Proof. Inhabitedness and roundedness of l are straightforward. For transitivity, suppose $q < l < r$, then we wish to show $q < r$. There exist $\varepsilon, \theta, \varepsilon', \theta' : \mathbb{Q}_{+}$ with $q + \varepsilon + \theta < x_{M(\varepsilon)}$ and $x_{M(\varepsilon')} < r - \varepsilon' - \theta'$. Now $\left|x_{M(\varepsilon)} - x_{M(\varepsilon')}\right| \leq \max(\varepsilon, \varepsilon')$, so either $q + \theta < x_{M(\varepsilon')}$ or $x_{M(\varepsilon)} < r - \theta$, and in either case $q < r$.

For locatedness, suppose $q < r$. Set $\varepsilon :\equiv \frac{r-q}{5}$, so that $q + 2\varepsilon < r - 2\varepsilon$. By locatedness of x_{ε}, we have $(q + 2\varepsilon < x_{\varepsilon}) \vee (x_{\varepsilon} < r - 2\varepsilon)$, hence $(q < l) \vee (l < r)$.

In order to show convergence, let $\varepsilon : \mathbb{Q}_{+}$, set $N :\equiv M(\varepsilon)$, and let $n \geq N$. We need to show $|x_n - l| \leq \varepsilon$, or equivalently, $-\varepsilon \leq x_n - l \leq \varepsilon$. For $x_n - l \leq \varepsilon$, suppose that $\varepsilon < x_n - l$, or equivalently, $l < x_n - \varepsilon$. There exist $\varepsilon', \theta' : \mathbb{Q}_{+}$ with $x_{M(\varepsilon')} < x_n - \varepsilon - \varepsilon' - \theta'$, or equivalently, $\varepsilon + \varepsilon' + \theta' < x_n - x_{M(\varepsilon')}$, which contradicts M being a modulus of Cauchy convergence. We can similarly show $-\varepsilon \leq x_n - l$. □

Proposition 4.6.7. *We have canonical inclusions $\mathbb{R}_{\mathbf{C}} \subseteq \mathbb{R}_{\mathbf{H}} \subseteq \mathbb{R}_{\mathbf{D}}$.*

Proof. The Cauchy reals are included into the HoTT book reals by Cauchy completeness of $\mathbb{R}_{\mathbf{H}}$: the equivalence class of a sequence–modulus pair (x, M) is simply included as the limit, computed in $\mathbb{R}_{\mathbf{H}}$, of that sequence of rationals.

The Univalent Foundations Program [91, Theorem 11.3.50] constructs $i_H : \mathbb{R}_{\mathbf{H}} \to \mathbb{R}_{\mathbf{D}}$ as a map which is both an embedding and a Cauchy structure morphism. □

4.7 Notes

The results in this chapter are essentially known, although we have attempted to optimize the arguments. For instance, Definition 4.1.10 captures the compatibility of the field operators with the field ordering with two laws, with a corresponding new lemma, namely Lemma 4.1.11, of equivalence with The Univalent Foundations Program [91].

The inclusion of the rationals into ordered fields, Lemma 4.3.3, was originally written to phrase the Archimedean property precisely. As far as we are aware, the observation that $<$ being preserved implies $<$ being reflected is new. This lemma allowed us to relate Cauchy structures and Cauchy complete Archimedean ordered fields in Corollary 4.5.11, and this important observation will be used again in Proposition 5.1.1.

The Cauchy reals, which we defined in Section 4.4, is a simple translation of well-known traditional definitions of the Cauchy reals. In UTT, more attention is paid to the HoTT book reals; however, the Cauchy reals will pop up naturally in Theorem 6.10.3.

The definition of the HoTT book reals as a homotopy-initial Cauchy structure is new, and will allow for certain neat new theorems and proofs in Chapter 5.

The main new element in our discussion of both-sided Dedekind cuts was the overloading of inequality relation, which enables writing concise definitions of Dedekind cuts.

Chapter 5

UNIVERSAL PROPERTIES OF REAL NUMBERS

Escardó and Simpson introduced the notion of *interval object*, which can be defined in any category with finite products, as a universal property for closed and bounded real line segments [43]. Indeed, in the category of classical sets, the real interval $[-1, 1]$ is an interval object. In the category of topological spaces, the real interval $[-1, 1]$ with the Euclidean topology is an interval object. Vickers [97] showed that in the category of locales, the locale corresponding to the interval $[-1, 1]$ is an interval object.

In a topos, the interval $[-1, 1]$ in a certain subobject $\mathbb{R}_\mathbf{E}$ of the Dedekind reals is an interval object. The object $\mathbb{R}_\mathbf{E}$, referred to as the Euclidean reals, is defined as the least Cauchy complete subset of the Dedekind reals containing the rationals. This can be constructed as the intersection of all Cauchy complete subsets of the Dedekind reals that contain the rationals.

Assuming the propositional resizing axiom of Definition 2.6.13, we can translate the construction of the Euclidean reals $\mathbb{R}_\mathbf{E}$ as an intersection of subsets of $\mathbb{R}_\mathbf{D}$ into type theory, and similarly translate the proof that the interval in $\mathbb{R}_\mathbf{E}$ is an interval object. The fact that $\mathbb{R}_\mathbf{E}$ is the least Cauchy complete subset of $\mathbb{R}_\mathbf{D}$ containing the rationals is then easily verified. The Euclidean reals sit between the Cauchy reals and the Dedekind reals: we have the sequence of canonical inclusions

$$\mathbb{R}_\mathbf{C} \subseteq \mathbb{R}_\mathbf{E} \subseteq \mathbb{R}_\mathbf{D}$$

where neither of the inclusions can be shown to be an equality. This reminds us of the HoTT

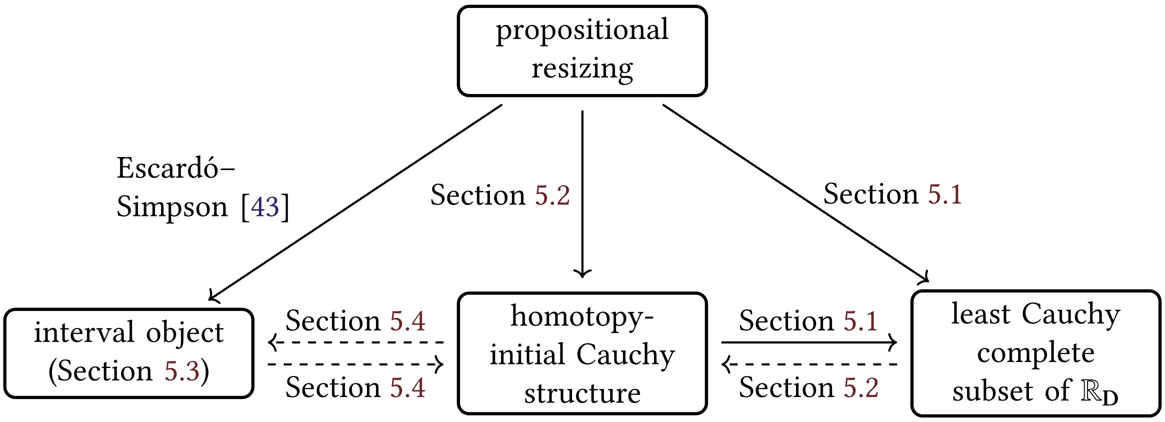

propositional resizing
Escardó–Simpson [43]
Section 5.2
Section 5.1
interval object (Section 5.3)
Section 5.4
Section 5.4
homotopy-initial Cauchy structure
Section 5.1
Section 5.2
least Cauchy complete subset of $\mathbb{R}_D$

Figure 5.1: Overview of relations

book reals of Section 4.5, which also sits between $\mathbb{R}_C$ and $\mathbb{R}_D$ in a canonical way:

$$\mathbb{R}_C \subseteq \mathbb{R}_H \subseteq \mathbb{R}_D.$$

This raises the question whether the HoTT book reals and the Euclidean reals coincide, so that the interval in the HoTT book reals would be an interval object, defined precisely in Section 5.3. The Univalent Foundations Program [91, Chapter 11, Notes] indeed conjectures that $\mathbb{R}_H$ and $\mathbb{R}_E$ coincide.

When phrasing this question more precisely, we are reminded that we may be working in a type theory in which we do not have the HoTT book reals, or in a type theory which does not have propositional resizing, so that we cannot define the Euclidean reals. In this chapter, we relate $\mathbb{R}_H$, $\mathbb{R}_E$ and interval objects in three different ways, sometimes assuming the existence of the HoTT book reals, and sometimes assuming propositional resizing, as visualized in Figure 5.1. Each box in the figure represents a certain assumption, and the arrows are logical implications. For example, assuming propositional resizing, it would be possible to define $\mathbb{R}_E$ and to repeat the result of Escardó–Simpson that the interval in $\mathbb{R}_E$ is an interval object.

We can relate $\mathbb{R}_H$ to $\mathbb{R}_E$ by showing that $\mathbb{R}_H$ is the least Cauchy complete subset of $\mathbb{R}_D$ containing the rationals, as we do in Section 5.1. This result can be phrased without propositional resizing, since $\mathbb{R}_E$ can be characterized as the least Cauchy complete subset of the Dedekind reals containing the rationals. In particular, when we do have propositional resizing, we can define $\mathbb{R}_E$ and, by this result, it coincides with $\mathbb{R}_H$. A previous version of this development [26] assumed both propositional resizing and the existence of $\mathbb{R}_{\mathbf{H}}$.

If we do not have $\mathbb{R}_{\mathbf{H}}$, we can still relate $\mathbb{R}_{\mathbf{E}}$ to $\mathbb{R}_{\mathbf{H}}$ by showing that $\mathbb{R}_{\mathbf{E}}$ satisfies a universal property similar to the one of the HoTT book reals given in Definition 4.5.12. This result in Section 5.2 assumes propositional resizing. In particular, when we do have $\mathbb{R}_{\mathbf{H}}$, this implies that $\mathbb{R}_{\mathbf{E}}$ and $\mathbb{R}_{\mathbf{H}}$ coincide.

We use propositional resizing to define $\mathbb{R}_{\mathbf{E}}$, and by the above, we can prove it has a certain universal property. We may also wonder whether a least Cauchy complete subset of the Dedekind reals containing the rationals, without knowing its construction as $\mathbb{R}_{\mathbf{E}}$, has this universal property; see Discussion 5.2.3. This question can be phrased without propositional resizing. Another open question is whether we can define a homotopy-initial Cauchy structure from an interval object, and vice-versa; see Section 5.4.

In summary, we confirm the conjecture of The Univalent Foundations Program [91] in two ways, once assuming the existence of $\mathbb{R}_{\mathbf{H}}$, and once assuming propositional resizing. We leave three relations, shown as dashed arrows in Figure 5.1, as open questions.

5.1 Subsets of the Dedekind reals

> The fact that $\mathbb{R}_c$ is the least Cauchy complete archimedean ordered field, as was proved in Theorem 11.3.50, indicates that our Cauchy reals probably coincide with the Escardó-Simpson reals. It would be interesting to check whether this is really the case.
>
> — *The Univalent Foundations Program [91, Chapter 11, Notes]. Note that we introduced this object as the "HoTT book reals, $\mathbb{R}_{\mathbf{H}}$" in Definition 4.5.12.*

In order to relate $\mathbb{R}_{\mathbf{H}}$ to $\mathbb{R}_{\mathbf{E}}$, without assuming propositional resizing, we relate $\mathbb{R}_{\mathbf{H}}$ to an arbitrary Cauchy complete subset R of $\mathbb{R}_{\mathbf{D}}$ that contains the rationals, using the homotopy-initiality of $\mathbb{R}_{\mathbf{H}}$ as in Definition 4.5.12. Theorem 11.3.50 in The Univalent Foundations Program [91] gives a canonical embedding of $\mathbb{R}_{\mathbf{H}}$ into R, which we redevelop in more detail in Section 5.1.1. So we reduce the question of coincidence of $\mathbb{R}_{\mathbf{H}}$ and $\mathbb{R}_{\mathbf{E}}$ to the fact that both are minimal Cauchy complete subsets of the Dedekind reals, answering the above question positively.

Conversely, in Section 5.1.2 we relate $\mathbb{R}_\mathrm{H}$ to $\mathbb{R}_\mathrm{E}$, without assuming $\mathbb{R}_\mathrm{H}$ exists, by showing that $\mathbb{R}_\mathrm{E}$ satisfies a homotopy-initiality property similar to that of $\mathbb{R}_\mathrm{H}$. This reduces the coincidence of $\mathbb{R}_\mathrm{H}$ and $\mathbb{R}_\mathrm{E}$ to uniqueness up to unique isomorphism of objects defined by a universal property.

In the particular case that we have both $\mathbb{R}_\mathrm{H}$ and propositional resizing, using either of these two developments, $\mathbb{R}_\mathrm{H}$ and $\mathbb{R}_\mathrm{E}$ coincide.

5.1.1 Minimality of the HoTT book reals

Let $R : \mathcal{P}\mathbb{R}_\mathrm{D}$ be a subtype of the Dedekind reals. We can consider the collection $(\Sigma x : \mathbb{R}_\mathrm{D})x \in R$ of elements in R, as in Section 2.6. We restrict the Cauchy structure of $\mathbb{R}_\mathrm{D}$ obtained from Corollary 4.5.11 to R.

Proposition 5.1.1. *Given a Cauchy complete subset $R : \mathcal{P}\mathbb{R}_\mathrm{D}$ of the Dedekind reals containing the rationals, the Cauchy structure on $\mathbb{R}_\mathrm{D}$ restricts to a Cauchy structure on R.*

Proof. First, the premetric on R is inherited from the one on $\mathbb{R}_\mathrm{D}$ by restriction: for $\varepsilon : \mathbb{Q}_+$ and $x, y : \mathbb{R}_\mathrm{D}$ with $\mu : x \in R$ and $\nu : x \in R$, we simply say that $(x, \mu) \sim_\varepsilon (y, \nu)$ holds iff $x \sim_\varepsilon y$.

The map $\mathrm{rat} : \mathbb{Q} \to \mathbb{R}_\mathrm{D}$ obtained via Corollary 4.5.11 from Lemma 4.3.3 is an embedding, so that we may see $\mathbb{Q}$ as a subtype $\mathbb{Q} : \mathcal{P}\mathbb{R}_\mathrm{D}$ of the Dedekind reals. Assuming R is a subtype of $\mathbb{R}_\mathrm{D}$ containing the rationals, i.e. $\mathbb{Q} \subseteq R \subseteq \mathbb{R}_\mathrm{D}$, we also get $\mathrm{rat} : \mathbb{Q} \to R$ by a straightforward restriction.

In order to phrase when we have a lim structure, we define a subset C_R of the type $C_{\mathbb{R}_\mathrm{D}}$, consisting of Cauchy approximations in R, by, for $x : C_{\mathbb{R}_\mathrm{D}}$,

$$C_R(x) :\equiv (\forall \varepsilon : \mathbb{Q}_+) x_\varepsilon \in R,$$

noting that this C_R, now seen as a type that embeds into $C_{\mathbb{R}_\mathrm{D}}$, is equivalent to the type of Cauchy approximations in $(\Sigma x : \mathbb{R}_\mathrm{D})x \in R$. By further assuming that R is Cauchy complete in the sense that for every Cauchy approximation $x \in C_R$ of elements in R, i.e. $x : C_{\mathbb{R}_\mathrm{D}}$ such that

$(\forall \varepsilon : \mathbb{Q}_+) x_\varepsilon \in R$, there exists a limit of x in R, we obtain a lim map: after all, we can compute the limit in $\mathbb{R}_\mathbf{D}$ using the lim structure of $\mathbb{R}_\mathbf{D}$, and then Cauchy completeness of R states that this unique limit is an element of R.

The construction of eq follows from (2.2) in Definition 2.6.2. That is, the projection map $\mathrm{pr}_0 : ((\Sigma x : \mathbb{R}_\mathbf{D}) x \in R) \to \mathbb{R}_\mathbf{D}$ is an equivalence between identity types of R and identity types of $\mathbb{R}_\mathbf{D}$, so that we may appeal to the eq structure of $\mathbb{R}_\mathbf{D}$.

The distance laws hold because the premetric on R is just the restriction of the premetric on $\mathbb{R}_\mathbf{D}$. □

Corollary 5.1.2. *The map that includes R into $\mathbb{R}_\mathbf{D}$ is a Cauchy structure morphism.*

Proposition 4.6.7 established $\mathbb{R}_\mathbf{H}$ as a subset of $\mathbb{R}_\mathbf{D}$ using a Cauchy structure morphism $i_H : \mathbb{R}_\mathbf{H} \to \mathbb{R}_\mathbf{D}$. So we have two subsets $\mathbb{R}_\mathbf{H}$ and R of $\mathbb{R}_\mathbf{D}$. The following proposition tells us that $\mathbb{R}_\mathbf{H} \subseteq R$.

Proposition 5.1.3. *We have $\mathbb{R}_\mathbf{H} \subseteq R$ as subsets of $\mathbb{R}_\mathbf{D}$. That is, there is a horizontal map in the following diagram making the triangle commute.*

$$\begin{tikzcd}
\mathbb{R}_\mathbf{H} \arrow[rr, dashed, "f"] \arrow[dr, hook, "i_H"'] \arrow[rr, phantom, shift right=6ex, "\bigcirc"] & & R \arrow[dl, hook', "i_R"] \\
& \mathbb{R}_\mathbf{D} &
\end{tikzcd}$$

Proof. By homotopy-initiality of the HoTT book reals, we obtain $f : \mathbb{R}_\mathbf{H} \to R$, and by the fact that Cauchy structure morphisms are closed under composition, using homotopy-initiality of the Cauchy structure of $\mathbb{R}_\mathbf{H}$ once more, we obtain the commutativity condition $i_R \circ f = i_H$. □

Lemma 2.6.9 tells us that the map $f : \mathbb{R}_\mathbf{H} \to R$ above is an embedding.

For an arbitrary Cauchy complete subset R of $\mathbb{R}_\mathbf{D}$ containing the rationals, we have shown that $\mathbb{R}_\mathbf{H} \subseteq R$. The HoTT book reals are the least Cauchy complete subset of the Dedekind reals containing the rationals.

Corollary 5.1.4. *Assuming $\mathbb{R}_\mathrm{H}$ exists, $\mathbb{R}_\mathrm{C}$ of Section 4.4 is Cauchy Complete iff $\mathbb{R}_\mathrm{C}$ and $\mathbb{R}_\mathrm{H}$ are the same subset of $\mathbb{R}_\mathrm{D}$.*

Proof. If $\mathbb{R}_\mathrm{C}$ is Cauchy complete, then it satisfies the conditions we set for R above. Conversely, if $\mathbb{R}_\mathrm{C}$ and $\mathbb{R}_\mathrm{H}$ coincide, then Cauchy completeness of $\mathbb{R}_\mathrm{H}$ also applies to $\mathbb{R}_\mathrm{C}$. □

5.1.2 Euclidean reals and interval objects

Escardó and Simpson [43] showed that, in any elementary topos, the Euclidean real interval is an interval object. They carried out the proof in a type theory for toposes [65, 72, 56], higher-order intuitionistic logic, which we adapt to our type theory, assuming propositional resizing.

Definition 5.1.5 (Escardó and Simpson [43])**.** Assuming propositional resizing, the type $\mathbb{R}_\mathrm{E}$ of *Euclidean reals* is defined as the meet (as in Lemma 2.6.15) of the subtypes of the Dedekind reals which are Cauchy complete and contain the rationals.

Discussion 5.1.6. We now take a moment to consider universe levels. Let $\mathbb{Q} : \mathcal{U}_i$, and recall from Section 4.6 that this means $\mathbb{R}_\mathrm{D} : \mathcal{U}_{i+1}$. For simplicity, let subsets R of $\mathbb{R}_\mathrm{D}$ correspond to functions $R : \mathcal{P}_i\mathbb{R}_\mathrm{D}$ that maps an element of $\mathbb{R}_\mathrm{D}$ to a proposition in $\mathcal{U}_i$. The condition $(\forall q : \mathbb{Q})i(q) \in R$ that R contains the rationals is canonically a proposition in HProp_i, and that it is Cauchy complete is canonically stated as the proposition $(\forall x : \mathcal{C}_{\mathbb{R}_\mathrm{D}})((\forall \varepsilon : \mathbb{Q}_+)x_\varepsilon \in R) \Rightarrow \lim x \in R$ in HProp_{i+1}. In other words, $\mathbb{R}_\mathrm{E}$ is the meet of a subset $\mathcal{P}_{i+1}\mathcal{P}_i\mathbb{R}_\mathrm{D}$. Hence, corresponding to Lemma 2.6.12, $\mathbb{R}_\mathrm{E}$, as a subset of $\mathbb{R}_\mathrm{D}$, is a function $\mathcal{P}_{i+1}\mathbb{R}_\mathrm{D}$: for a given $x : \mathbb{R}_\mathrm{D}$, it outputs a proposition that quantifies over all $R : \mathcal{P}_i\mathbb{R}_\mathrm{D}$ that satisfy the condition in HProp_{i+1} that R is Cauchy complete and contains the rationals.

Using propositional resizing, $\mathbb{R}_\mathrm{E}$ can also be seen as an element of $\mathcal{P}_i\mathbb{R}_\mathrm{D}$.

The definition of $\mathbb{R}_\mathrm{E}$ as an intersection of Cauchy complete subsets of $\mathbb{R}_\mathrm{D}$ containing the rationals makes $\mathbb{R}_\mathrm{E}$ itself into the least such subset of $\mathbb{R}_\mathrm{D}$.

Theorem 5.1.7. *Assuming propositional resizing and that $\mathbb{R}_\mathrm{H}$ exists, $\mathbb{R}_\mathrm{E}$ coincides with $\mathbb{R}_\mathrm{H}$ as subsets of $\mathbb{R}_\mathrm{D}$.*

Proof. The fact that $\mathbb{R}_\mathrm{H} \subseteq \mathbb{R}_\mathrm{E}$ is Proposition 5.1.3. Since $\mathbb{R}_\mathrm{H}$ is a Cauchy complete subset of the Dedekind reals containing the rationals, we have $\mathbb{R}_\mathrm{E} \subseteq \mathbb{R}_\mathrm{H}$. Hence $\mathbb{R}_\mathrm{E} = \mathbb{R}_\mathrm{H}$. □

From Theorem 10.1.12 in The Univalent Foundations Program [91], we know that the type of all sets in a given universe is a topos. This allows us to interpret Escardó and Simpson's definition, and construction, of interval objects in toposes.

Theorem 5.1.8. *Assuming propositional resizing, so that we can construct $\mathbb{R}_\mathrm{E}$ as an element of some universe $\mathcal{U}$. The unit interval in $\mathbb{R}_\mathrm{E}$ is an interval object, where interval objects are defined as in Escardó and Simpson with respect to that category of sets in universe $\mathcal{U}$.*

Proof. This is simply a translation of the proof in Escardó and Simpson [43], where we note that our definition of $\mathbb{R}_\mathrm{E}$ coincides with the definition in category-theoretic terms. □

Since the HoTT book reals are a set [91, Theorem 11.3.9], as a consequence, the interval in the HoTT book reals is an interval object.

Corollary 5.1.9. *Assuming propositional resizing, so that we can construct $\mathbb{R}_\mathrm{E}$ as an element of some universe $\mathcal{U}$, and that $\mathbb{R}_\mathrm{H}$ exists in $\mathcal{U}$. The unit interval in $\mathbb{R}_\mathrm{H}$ is an interval object, where interval objects are defined as in Escardó and Simpson with respect to that category of sets in universe $\mathcal{U}$.*

Instead of this approach, we discuss the possibility of a direct proof that the interval in $\mathbb{R}_\mathrm{H}$ is an interval object in Section 5.4, avoiding propositional resizing.

5.2 Homotopy-initiality of the Euclidean reals

In the previous section, we have related $\mathbb{R}_\mathrm{H}$ and $\mathbb{R}_\mathrm{E}$ by showing that $\mathbb{R}_\mathrm{H}$ is the least Cauchy complete subset of $\mathbb{R}_\mathrm{D}$ containing the rationals—a result that requires having the type $\mathbb{R}_\mathrm{H}$ in the first place. In a type theory where $\mathbb{R}_\mathrm{H}$ is not given as a primitive type, we can still relate the Euclidean reals and the HoTT book reals. The HoTT book reals are defined uniquely by their universal property; that is, any two homotopy-initial Cauchy structures are equal. The goal of this section is to show that $\mathbb{R}_\mathbf{E}$ satisfies that same universal property, so that when we do have $\mathbb{R}_\mathbf{H}$, it coincides with $\mathbb{R}_\mathbf{E}$.

We borrow two strategies from the proof of Escardó–Simpson [43, draft full version] that the interval in the Euclidean reals is an interval object, namely

1. defining a dcpo, such that the construction of a certain point of that dcpo corresponds to proving the theorem, and
2. using a fixed point theorem, based on Pataraia's [79], to construct that point.

Concretely, we need to show that for any Cauchy structure $(S, \sim)$, the type CS-hom$(\mathbb{R}_\mathbf{E}, S)$ of Cauchy structure morphisms is contractible. So for a given Cauchy structure $(S, \sim)$, we define a certain subdcpo $\mathcal{F}_{(S,\sim)}$ of $\mathcal{P}_{i+1}\mathbb{R}_\mathbf{D}$ whose elements are subsets $\mathbb{Q} \subseteq R \subseteq \mathbb{R}_\mathbf{E}$ for which, loosely speaking, the type of Cauchy structure morphisms *restricted to* R is contractible. By showing that $\mathbb{R}_\mathbf{E}$ is an element of $\mathcal{F}_{(S,\sim)}$, we have the required result. In particular, $\mathbb{R}_\mathbf{E}$ is found as a fixed point of a certain $\mathcal{F}_{(S,\sim)}$-closed endomap Φ, which extends a subset R to the set of limits of sequences valued in R.

The definitions of $\mathcal{F}_{(S,\sim)}$ and Φ loosely follow the style of Escardó–Simpson, but have some changes since we are showing a different universal property and working in a different logic.

The construction of $\mathcal{F}_{(S,\sim)}$ and Φ, and establishing their required properties, requires extensive calculations, since the construction of an element of $\mathcal{F}_{(S,\sim)}$ requires showing that a certain type of restricted Cauchy structure morphisms is contractible. This contractibility, in turn, consists of the construction of a restricted Cauchy structure morphism, and a proof of uniqueness of those restricted Cauchy structure morphisms. The fact that the fixed point theorem that we use has weaker assumptions than, for instance, Kleene's or Knaster–Tarski's works to our advantage.

Although the proof of Pataraia's fixed point theorem uses the propositional resizing axiom of Definition 2.6.13, we use Corollary 3.2.2, which does not require it. However, we do use propositional resizing to appeal to Lemma 3.1.7, which gives that $\mathcal{P}_{i+1}\mathbb{R}_{\mathbf{D}}$ is a dcpo. We use the specific construction of the joins in our proof that $\mathcal{F}_{(S,\sim)}$ is a subdcpo.

Theorem 5.2.1. *Assuming propositional resizing, the Euclidean reals satisfy the universal property of the HoTT book reals of Section 4.5 for sets. That is, for a Cauchy structure $(S, \sim)$, where S is a set, the type* $\text{CS-hom}(\mathbb{R}_{\mathbf{E}}, S)$ *of Cauchy structure morphisms from $\mathbb{R}_{\mathbf{E}}$ to S is contractible.*

> *Remark* 5.2.2. Since S is a set, saying that $\text{CS-hom}(\mathbb{R}_{\mathbf{E}}, S)$ is contractible is equivalent to saying that there exists a Cauchy structure morphism from $\mathbb{R}_{\mathbf{E}}$ to S, and any two such morphisms are pointwise equal.

It would be desirable to be able to prove homotopy-initiality for arbitrary types S, rather than only for sets, but we leave this as an open problem. A similar issue arises in work by Awodey, Frey and Speight on impredicative encodings of higher inductive types [8].

We refer to the data of the Cauchy structure on $\mathbb{R}_{\mathbf{D}}$ as rat, lim and eq, and to the data of another Cauchy structure $(S, \sim)$ with subscripts as rat_S, lim_S and eq_S.

Throughout this section, we make extensive usage of the notation for subsets as in Section 2.6, and in particular the notation for quantification over subtypes introduced in Definition 2.6.16.

Recall from Discussions 4.6.4 and 5.1.6 that if we start with some type of rationals $\mathbb{Q}$ in universe i, then we have $\mathbb{R}_{\mathbf{D}}, \mathbb{R}_{\mathbf{E}} : \mathcal{U}_{i+1}$.

Proof. As in Section 5.1.1, for a given subset $Y : \mathcal{P}_{i+1}\mathbb{R}_{\mathbf{D}}$ of the Dedekind reals, we define a subset $C_Y : \mathcal{P}_{i+1}C_{\mathbb{R}_{\mathbf{D}}}$ of the type $C_{\mathbb{R}_{\mathbf{D}}}$ of Cauchy approximations in $\mathbb{R}_{\mathbf{D}}$, with $x : C_{\mathbb{R}_{\mathbf{D}}}$, as

$$C_Y(x) :\equiv (\forall \varepsilon : \mathbb{Q}_+) x_\varepsilon \in Y.$$

For a subset Y of $\mathbb{R}_{\mathbf{D}}$ with $\mathbb{Q} \subseteq Y$, we define what it means to have a restricted Cauchy structure morphism $Y \to S$. Compared to ordinary Cauchy structure morphism as in Definition 4.5.8, the essence of the definition is that although the output of $\text{rat} : \mathbb{Q} \to \mathbb{R}_{\mathbf{D}}$ is always an element of Y, because $\mathbb{Q} \subseteq Y$, the output of $\lim : C_Y \to \mathbb{R}_{\mathbf{D}}$ may not be, and so we require the corresponding preservation condition for Y only in the case that it is. Additionally, because S is a set, preservation of the eq structure is automatic. In conclusion, we define

$$\begin{aligned}\text{sub-CS-hom}(Y,S) :=& (\Sigma f : Y \to S)\\ &(\Sigma g : (\Pi \varepsilon : \mathbb{Q}_+)(\Pi u, v \in Y) u \sim_\varepsilon v \to f(u) \sim_\varepsilon f(v))\\ &((\Pi q : \mathbb{Q}) f(\mathrm{rat}(q)) = \mathrm{rat}_S(q))\\ &\times((\Pi x \in C_Y) \lim x \in Y \Rightarrow f(\lim x) = \lim_S(f \circ x))\end{aligned}$$

where, following Definition 2.6.16, $(\Pi u, v \in Y)C(u,v)$ means $(\Pi u, v : \mathbb{R}_\mathbf{D}) u, v \in Y \Rightarrow C(u,v)$, and similarly $(\Pi x \in C_Y)D(x)$ means $(\Pi x : C_{\mathbb{R}_\mathbf{D}})x \in C_Y \Rightarrow D(x)$.

Note that $\text{sub-CS-hom}(\mathbb{R}_\mathbf{E}, S) \simeq \text{CS-hom}(\mathbb{R}_\mathbf{E}, S)$ because lim is always defined on $\mathbb{R}_\mathbf{E}$. The goal is to show that $\mathbb{R}_\mathbf{E}$ is an element of the subset $\mathcal{F}_{(S,\sim)} : \mathcal{P}_{i+1}\mathcal{P}_{i+1}\mathbb{R}_\mathbf{D}$ of $\mathcal{P}_{i+1}\mathbb{R}_\mathbf{D}$ defined by

$$\mathcal{F}_{(S,\sim)}(Y) := \mathbb{Q} \subseteq Y \subseteq \mathbb{R}_\mathbf{E} \wedge \text{isContr}(\text{sub-CS-hom}(Y,S)),$$

so that there is a unique Cauchy structure morphism from $\mathbb{R}_\mathbf{E}$ to S. We show this by using Corollary 3.2.2 to construct a fixed point of a certain map Φ that we will define later, and then showing that this fixed point is a Cauchy complete subset of $\mathbb{R}_\mathbf{E}$, so that it must be equal to $\mathbb{R}_\mathbf{E}$.

Note that two restricted Cauchy structure morphisms, that is, two elements of the type $\text{sub-CS-hom}(Y,S)$, are equal iff their underlying maps $Y \to S$ are equal, because the remaining data is a proposition.

First, to be more precise, in order to be able to use Corollary 3.2.2, we show the following claims.

Claim 1. $\mathcal{P}_{i+1}\mathbb{R}_\mathbf{D}$ is an $(i+2)$-dcpo with the relation $\subseteq$.

Claim 2. $\mathcal{F}_{(S,\sim)}$ is a subdcpo of $\mathcal{P}_{i+1}\mathbb{R}_\mathbf{D}$.

Claim 3. $\mathbb{Q} \in \mathcal{F}_{(S,\sim)}$.

Claim 4. The map $\Phi : \mathcal{P}_{i+1}\mathbb{R}_\mathbf{D} \to \mathcal{P}_{i+1}\mathbb{R}_\mathbf{D}$, which we define later, is inflationary.

Claim 5. $\mathcal{F}_{(S,\sim)}$ is Φ-closed.

Proof of Claim 1. By Lemma 3.1.7, indeed $\mathcal{P}_{i+1}\mathbb{R}_{\mathrm{D}}$ is an $(i+2)$-dcpo. □

Proof of Claim 2. To show that $\mathcal{F}_{(S,\sim)}$ is an $(i+2)$-subdcpo, let $\mathcal{D} : \mathcal{P}_{i+2}\mathcal{P}_{i+1}\mathbb{R}_{\mathrm{D}}$ with $\mathcal{D} \subseteq \mathcal{F}_{(S,\sim)}$ be a directed subset of $\mathcal{F}_{(S,\sim)}$. Following Corollary 2.6.15, the join of $\mathcal{D}$ in $\mathcal{P}_{i+1}\mathbb{R}_{\mathrm{D}}$ is constructed using propositional resizing as $Y = \bigcup \mathcal{D}$ with $Y : \mathcal{P}_{i+1}\mathbb{R}_{\mathrm{D}}$, and we claim that it is an element of $\mathcal{F}_{(S,\sim)}$. Because $\mathcal{D} \subseteq \mathcal{F}_{(S,\sim)}$, the various elements $X \in \mathcal{D}$ give rise to their own restricted Cauchy structure morphism which is unique on X, and we refer to their underlying maps as $f_X : X \to S$ and $g_X : (\Pi\varepsilon : \mathbb{Q}_+)(\Pi u, v \in X) u \sim_\varepsilon v \to f_X(u) \sim_\varepsilon f_X(v)$.

$\mathbb{Q} \subseteq Y \subseteq \mathbb{R}_{\mathrm{E}}$ holds because the elements of $\mathcal{D}$ satisfy this property.

To show that sub-CS-hom(Y, S) is a proposition, consider two restricted Cauchy structure morphisms with maps $f, f' : Y \to S$, and let $y \in Y$, recalling from Definition 2.6.16 that this means we take $y : \mathbb{R}_{\mathrm{D}}$ and assume $y \in Y$. We aim to show the proposition $f(y) = f'(y)$, so we may assume to have $X \in \mathcal{D}$ with $y \in X$. Both f and f' restrict to restricted Cauchy structure morphisms on X, where they must both equal the center of contraction given by $f_X : X \to S$, and in particular $f(y) = f_X(y) = f'(y)$.

We construct $f_Y : Y \to S$ as a certain map

$$f'_Y : (\Pi y \in Y)(\Sigma s : S)(\exists X \in \mathcal{D}) y \in X \wedge f_X(y) = s$$

composed with a projection map that forgets the proof of $(\exists X \in \mathcal{D}) y \in X \wedge f_X(y) = s$. Notice that for every y, the codomain $(\Sigma s : S)(\exists X \in \mathcal{D}) y \in X \wedge f_X(y) = s$ of f'_Y is a proposition, because given $s, s' : S$ and $X, X' \in \mathcal{D}$ with $y \in X$ and $y \in X'$ and $f_X(y) = s$ and $f_{X'}(y) = s'$, from the fact that $\mathcal{D}$ is directed, we know that there exists $Z \in \mathcal{D}$ with $X, X' \subseteq Z$. But the map f_Z restricts to both X and X' where it must be equal to f_X and $f_{X'}$, respectively, so that $s = f_X(y) = f_Z(y) = f_{X'}(y) = s'$.

To construct f'_Y, take an element $y \in Y$. By the construction of $Y = \bigcup \mathcal{D}$, this means there exists $X \in \mathcal{D}$ with $y \in X$. Since the codomain is a proposition, we may assume to *have* $X \in \mathcal{D}$ with $y \in X$. Set $s :\equiv f_X(y)$.

To construct $g_Y : (\Pi\varepsilon : \mathbb{Q}_+)(\Pi u, v \in Y) u \sim_\varepsilon v \to f_Y(u) \sim_\varepsilon f_Y(v)$, let $\varepsilon : \mathbb{Q}_+$, let $X, X' \in \mathcal{D}$ with $u \in X$ and $v \in X'$, and let $\nu : u \sim_\varepsilon v$. Because $\mathcal{D}$ is directed, we know that there exists

$Z \in \mathcal{D}$ with $X, X' \subseteq Z$, so that we can output $g_Z(\varepsilon, u, v, \nu)$.

To show the preservation conditions, first note that $(\forall X \in \mathcal{D})(\forall x \in X) f_Y(x) = f_X(x)$, because $f_Y(x) : S$ and $f_X(x) : S$ both arise as elements of the codomain $(\Sigma s : S)(\exists X \in \mathcal{D}) y \in X \wedge f_X(y) = s$ of f'_Y, which is a proposition, as shown above.

To show that $(\Pi q : \mathbb{Q}) f_Y(\mathrm{rat}(q)) = \mathrm{rat}_S(q)$, let $q : \mathbb{Q}$. Since $\mathcal{D}$ is inhabited, there exists $X \in \mathcal{D}$, and since we are showing a proposition, we may assume to have such an X. Then, because f_X satisfies the preservation conditions, $f_Y(\mathrm{rat}(q)) = f_X(\mathrm{rat}(q)) = \mathrm{rat}_S(q)$.

To show that $(\Pi x \in C_Y) \lim x \in Y \Rightarrow f_Y(\lim x) = \lim_S(f_Y \circ x)$, let $x \in C_Y$ and assume $\lim x \in Y$.

One may be inclined to look for $X \in \mathcal{D}$ with $x_\varepsilon \in X$ for all $\varepsilon : \mathbb{Q}_+$, and also $\lim x \in X$, suggesting that we need $\mathcal{D}$ to be infinitary-directed, meaning that we would have an element in $\mathcal{D}$ which contains all x_ε. In fact, we can avoid this by observing that $\lim x$ can be computed as the limit of the constant Cauchy approximation $\lambda\varepsilon'. \lim x$. If we can show that the Cauchy approximation $f_Y \circ x$ is close to the constant Cauchy approximation $\lambda\varepsilon'.f_Y(\lim x)$, then we can use eq_S to show the required preservation condition. We now make this argument more precise.

First, note that since $\mathbb{R}_\mathbf{D}$ is Cauchy complete indeed we have $\lim x = \lim(\lambda\varepsilon'. \lim x)$. Since $\lim x \in Y$, and since we are showing a proposition, we may assume to have $X \in \mathcal{D}$ with $\lim x \in X$. Then

$$\begin{aligned} f_Y(\lim x) &= f_X(\lim x) \\ &= f_X(\lim(\lambda\varepsilon'. \lim x)) \\ &= \lim_S(\lambda\varepsilon'.f_X(\lim x)) \\ &= \lim_S(\lambda\varepsilon'.f_Y(\lim x)). \end{aligned}$$

By eq_S, it suffices to show

$$(\forall \varepsilon : \mathbb{Q}_+) \lim_S(\lambda\varepsilon'.f_Y(\lim x)) \sim_\varepsilon \lim_S(f_Y \circ x).$$

Let $\varepsilon : \mathbb{Q}_+$. The fourth distance law of Cauchy structures gives us, with $\varepsilon/2$, $\varepsilon/4$ and $\varepsilon/4$

respectively for ε, δ and η:

$$f_Y(\lim x) \sim_{\varepsilon/2} f_Y(x_{\varepsilon/4}) \to \lim_S(\lambda\varepsilon'.f_Y(\lim x)) \sim_\varepsilon \lim_S(f_Y \circ x).$$

In order to show the proposition $f_Y(\lim x) \sim_{\varepsilon/2} f_Y(x_{\varepsilon/4})$, from directedness of $\mathcal{D}$ we obtain $X \in \mathcal{D}$ with $\lim x \in X$ and $x_{\varepsilon/4} \in X$. Then

$$g_X(\varepsilon/2, \lim x, x_{\varepsilon/4}) : \lim x \sim_{\varepsilon/2} x_{\varepsilon/4} \to f_X(\lim x) \sim_{\varepsilon/2} f_X(x_{\varepsilon/4}),$$

and $\lim x \sim_{\varepsilon/2} x_{\varepsilon/4}$ can be shown using Cauchy completeness (as in Definition 4.5.7) of $\mathbb{R}_\mathbf{D}$.

This concludes the proof of Claim 2 that $\mathcal{F}_{(S,\sim)}$ is an $(i+2)$-subdcpo of $\mathcal{P}_{i+1}\mathbb{R}_\mathbf{D}$. □

Proof of Claim 3. To show that $\mathbb{Q} \in \mathcal{F}_{(S,\sim)}$, note that $\mathbb{Q} \subseteq \mathbb{Q} \subseteq \mathbb{R}_\mathbf{E}$. To show that the type sub-CS-hom$(\mathbb{Q}, S)$ is a proposition, let $f, f' : \mathbb{Q} \to S$ be Cauchy structure morphisms. Since they both satisfy the preservation condition for rationals, we have $f(\mathrm{rat}(q)) = \mathrm{rat}_S(q) = f'(\mathrm{rat}(q))$, as required.

Now we construct an element of sub-CS-hom$(\mathbb{Q}, S)$. The map $f_\mathbb{Q} : \mathbb{Q} \to S$ is given by rat_S directly. Then $g_\mathbb{Q}$ can be constructed using the first distance law on the Cauchy structure $(S, \sim)$. The preservation condition for rationals holds by definition. Let $x \in C_\mathbb{Q}$ and assume $\lim x \in \mathbb{Q}$. We need to show $f(\lim x) = \lim_S(f \circ x)$, i.e. $\mathrm{rat}_S(\lim x) = \lim_S(f \circ x)$. So by eq_S and the second distance law it suffices to show for arbitrary $\varepsilon : \mathbb{Q}_+$ that $\mathrm{rat}_S(\lim x) \sim_{2\varepsilon/3} f(x_{\varepsilon/3})$, i.e. that $\mathrm{rat}_S(\lim x) \sim_{2\varepsilon/3} \mathrm{rat}(x_{\varepsilon/3})$, i.e. by the first distance law that $-2\varepsilon/3 < \lim x - x_{\varepsilon/3} < 2\varepsilon/3$, which holds because $\lim x$ is a limit of x. □

Proof of Claim 4. We now define an inflationary $\mathcal{F}_{(S,\sim)}$-closed map Φ whose fixed point we will show to be $\mathbb{R}_\mathbf{E}$.

For $X : \mathcal{P}_{i+1}\mathbb{R}_\mathbf{D}$, define $\Phi(X) : \mathcal{P}_{i+1}\mathbb{R}_\mathbf{D}$ to be the subset of $\mathbb{R}_\mathbf{D}$ of limits of Cauchy approximations valued in X, that is:

$$\Phi(X)(y) :\equiv (\exists x \in C_X) y = \lim x$$

The map Φ is increasing because every real is the limit of a constant sequence, and monotone because if $X \subseteq Y$ then $C_X \subseteq C_Y$. □

Proof of Claim 5. To show that Φ is $\mathcal{F}_{(S,\sim)}$-closed, assume $X \in \mathcal{F}_{(S,\sim)}$, and note that $\mathbb{Q} \subseteq \Phi(X)$ holds because $\mathrm{rat}(q) = \lim(\lambda\varepsilon'.\,\mathrm{rat}(q))$, and $\Phi(X) \subseteq \mathbb{R}_{\mathbf{E}}$ follows from Cauchy completeness of $\mathbb{R}_{\mathbf{E}}$.

From isContr(sub-CS-hom(X, S)) we obtain $f_X : X \to S$.

To show that sub-CS-hom$(\Phi(X), S)$ is a proposition, consider two restricted Cauchy structure morphisms with maps $f, f' : \Phi(X) \to S$, and let $x \in C_X$. To show the proposition $f(\lim x) = f'(\lim x)$, note that f and f' restrict to the same restricted Cauchy structure morphism f_X on X. Then, since f and f' satisfy the preservation condition for the limit of x, we have $f(\lim x) = \lim_S(f \circ x) = \lim_S(f' \circ x) = f'(\lim x)$.

We define $f_{\Phi(X)} : \Phi(X) \to S$ as a certain map

$$f'_{\Phi(X)} : (\Pi y \in \Phi(X))(\Sigma s : S)(\exists x \in C_X) y = \lim x \wedge s = \lim_S(f_X \circ x)$$

followed by a projection map that forgets the proof of $(\exists x \in C_X) s = \lim_S(f_X \circ x)$.

First we show that the codomain of $f'_{\Phi(X)}$ is a proposition. For suppose $s, s' : S$ and $x, x' \in C_X$ with $y = \lim x = \lim x'$ and $s = \lim_S(f_X \circ x)$ and $s' = \lim_S(f_X \circ x')$. Because lim computes limits in $\mathbb{R}_{\mathbf{D}}$, we know that $\lim x = \lim x'$ implies

$$(\forall \varepsilon, \varepsilon', \theta, \theta' : \mathbb{Q}_+) x_\varepsilon \sim_{\varepsilon+\varepsilon'+\theta+\theta'} x_{\varepsilon'}$$

and hence in particular

$$(\forall \varepsilon : \mathbb{Q}_+) x_{\varepsilon/6} \sim_{4\varepsilon/6} x'_{\varepsilon/6}.$$

Now g_X gives us

$$(\forall \varepsilon : \mathbb{Q}_+) f_X(x_{\varepsilon/6}) \sim_{4\varepsilon/6} f_X(x'_{\varepsilon/6})$$

and so by the fourth distance law of the Cauchy structure S

$$(\forall \varepsilon : \mathbb{Q}_+) \lim_S(f_X \circ x) \sim_\varepsilon \lim_S(f_X \circ x')$$

and so with the eq_S, we get $s = \lim_S(f_X \circ x) = \lim_S(f_X \circ x') = s'$, as required.

Since the codomain of $f'_{\Phi(X)}$ is a proposition, for a given $y \in \Phi(X)$ we may assume to have $x \in C_X$ with $y = \lim x$. Then we can compute the output as $\lim_S(f_X \circ x)$, completing the definition of $f'_{\Phi(X)}$ and $f_{\Phi(X)} : \Phi(X) \to S$.

To define

$$g_{\Phi(X)} : (\Pi \varepsilon : \mathbb{Q}_+)(\Pi u, v \in \Phi(X)) u \sim_\varepsilon v \to f_{\Phi(X)}(u) \sim_\varepsilon f_{\Phi(X)}(v),$$

let $\varepsilon : \mathbb{Q}_+$, $x, y \in C_X$ and $\nu : \lim x \sim_\varepsilon \lim y$. We aim to show $f_{\Phi(X)}(\lim x) \sim_\varepsilon f_{\Phi(X)}(\lim y)$, i.e. $\lim_S(f_X \circ x) \sim_\varepsilon \lim_S(f_X \circ y)$ by the above definition of $f_{\Phi(X)}$.

Since $\lim x \sim_\varepsilon \lim y$, that is, $|\lim x - \lim y| < \varepsilon$, by the Archimedean property we know that

$$(\exists \delta : \mathbb{Q}_+)\ |\lim x - \lim y| < \delta < \varepsilon.$$

Since we are showing a proposition, we may assume to have such a δ. Because lim computes limits in $\mathbb{R}_\mathbf{D}$, using the definition of $\sim$ in $\mathbb{R}_\mathbf{D}$ we know that $\lim x \sim_\delta \lim y$ implies

$$(\forall \xi, \xi', \theta, \theta' : \mathbb{Q}_+) x_\xi \sim_{\delta+\xi+\xi'+\theta+\theta'} y_{\xi'}$$

and so in particular with $\xi := \frac{\varepsilon-\delta}{6}$ we have $x_\xi \sim_{\delta+4\xi} y_\xi$. Then g_X gives $f_X(x_\xi) \sim_{\delta+4\xi} f_X(y_\xi)$ and hence by the fourth distance law $\lim_S(f_X \circ x) \sim_\varepsilon \lim_S(f_X \circ y)$.

To show that $f_{\Phi(X)}$ and $g_{\Phi(X)}$ satisfy the coherence conditions for restricted Cauchy structure morphisms, let $q : \mathbb{Q}$. Then $f_{\Phi(X)}(\mathrm{rat}(q))$ may be computed as $\lim_S(\lambda\varepsilon'.f_X(\mathrm{rat}(q)))$, which, by the fact that f_X is a restricted Cauchy structure morphism, is equal to $\lim_S(\lambda\varepsilon'.\,\mathrm{rat}_S(q))$. By eq_S it suffices to show

$$(\forall \varepsilon : \mathbb{Q}_+) \lim_S(\lambda\varepsilon'.\,\mathrm{rat}_S(q)) \sim_\varepsilon \mathrm{rat}_S(q),$$

so let $\varepsilon : \mathbb{Q}_+$. Then $\mathrm{rat}_S(q) \sim_{\varepsilon/2} \mathrm{rat}_S(q)$ by the first distance law, and so $\lim_S(\lambda\varepsilon'.\,\mathrm{rat}_S(q)) \sim_\varepsilon \mathrm{rat}_S(q)$ by the third distance law.

For the second preservation condition, let $x \in C_{\Phi(X)}$ and assume $\lim x \in \Phi(X)$, that is, $(\exists x' \in C_X) \lim x = \lim x'$. Since we are showing the proposition $f_{\Phi(X)}(\lim x) = \lim_S(f_{\Phi(X)} \circ x)$, let x' be such, so that we have to show $\lim_S(f_X \circ x') = \lim_S(f_{\Phi(X)} \circ x)$. By eq_S, it suffices to show for $\varepsilon : \mathbb{Q}_+$ that

$$\lim_S(f_X \circ x') \sim_\varepsilon \lim_S(f_{\Phi(X)} \circ x).$$

Using the fourth distance law, it suffices to show

$$f_X(x'_{\varepsilon/6}) \sim_{4\varepsilon/6} f_{\Phi(X)}(x_{\varepsilon/6}).$$

Now

$$f_X(x'_{\varepsilon/6}) = f_X(\lim(\lambda\varepsilon'.x'_{\varepsilon/6})) = \lim_S(f_X \circ (\lambda\varepsilon'.x'_{\varepsilon/6})) = f_{\Phi(X)}(\lambda\varepsilon'.x'_{\varepsilon/6}) = f_{\Phi(X)}(x'_{\varepsilon/6}),$$

so this is equivalent to

$$f_{\Phi(X)}(x'_{\varepsilon/6}) \sim_{4\varepsilon/6} f_{\Phi(X)}(x_{\varepsilon/6})$$

and so by $g_{\Phi(X)}$ it suffices to show

$$x'_{\varepsilon/6} \sim_{4\varepsilon/6} x_{\varepsilon/6}$$

which holds because $\lim x = \lim x'$.

This concludes the proof of Claim 5 that Φ is $\mathcal{F}_{(S,\sim)}$-closed. □

Hence, by Corollary 3.2.2, Φ has a fixed point R in $\mathcal{F}_{(S,\sim)}$. By definition, $R \subseteq \mathbb{R}_\mathbf{E}$. It remains to show that $\mathbb{R}_\mathbf{E} \subseteq R$, which will follow from the fact that R is a Cauchy complete subset of the Dedekind reals containing the rationals. Additionally, the fact that R contains the rationals is part of the definition of $\mathcal{F}_{(S,\sim)}$, so we only have to show that R is Cauchy complete.

Let $x \in C_R$. By definition, we have $\lim x \in \Phi(R)$. Since R is a fixed point of Φ, we have $\lim x \in R$, and this is a limit of x because it is its limit in $\mathbb{R}_\mathbf{D}$.

Hence $\mathbb{R}_\mathbf{E}$ is an element of $\mathcal{F}_{(S,\sim)}$. □

Discussion 5.2.3. Now we consider what happens in the absence of propositional resizing.

The construction of $\mathbb{R}_\mathbf{E}$ will not go through, as our construction of an intersection of subsets $\mathcal{P}_i\mathbb{R}_\mathbf{D}$ of $\mathbb{R}_\mathbf{D}$ results in a subset $\mathcal{P}_{i+1}\mathbb{R}_\mathbf{D}$, which is not a true meet because, living in the wrong universe, it has the wrong type.

But suppose given any Cauchy complete $\mathbb{R} : \mathcal{P}_i\mathbb{R}_\mathbf{D}$ containing the rationals, which is the least such subset, can we prove a homotopy-initiality theorem similar to Theorem 5.2.1, replacing instances of $\mathbb{R}_\mathbf{E}$ with $\mathbb{R}$? The fixed point theorem, Corollary 3.2.2, that we used in the proof of Theorem 5.2.1, does not use propositional resizing, and we also do not need it to construct the desired fixed point $\mathbb{R}$ since we simply assume it to be given. We cannot straightforwardly apply Corollary 3.2.2, since we cannot show $\mathcal{P}_{i+1}\mathbb{R}_{\mathbf{D}}$ to be a dcpo. It may suffice to see $\mathcal{P}_{i+1}\mathbb{R}_{\mathbf{D}}$ as a partial order, and $\mathcal{F}_{(S,\sim)}$ as a subdcpo of that partial order in the sense that it contains all the joins of directed subsets *that exist in* $\mathcal{P}_{i+1}\mathbb{R}_{\mathbf{D}}$. Showing that $\mathcal{F}_{(S,\sim)}$ is a subdcpo in this sense would still require the construction of a restricted Cauchy structure morphism with some underlying maps f_Y and g_Y for a join Y of a directed subset $\mathcal{D}$ as in the proof above. In the absence of propositional resizing, we can not construct Y using existential quantifiers as in Corollary 2.6.8, and so the construction of f_Y and g_Y in the proof of Theorem 5.2.1 will not go through.

5.3 Interval objects

Sections 5.1 and 5.2 state results that are related to interval objects. In order to state some open questions in Section 5.4, we now define interval objects in our type theory.

Such a type-theoretic definition is necessary since the definition by Escardó–Simpson [43] is not stated in the internal language of a topos, but in what type-theoretically would be considered the metatheory.

Escardó–Simpson start with a binary algebra in a category with binary products, namely a pair (A, m) consisting of an object A with a morphism $m : A \times A \to A$. Such a binary algebra is a *cancellative midpoint algebra* if it is idempotent, commutative, and satisfies a transposition and cancellation law. This is easily phrased in type theory.

Definition 5.3.1. A *cancellative midpoint algebra* is a set A with a map $m : A \times A \to A$ satisfying:

1. *idempotent*: $(\forall x : A) m(x, x) = x$,
2. *commutative*: $(\forall x, y : A) m(x, y) = m(y, x)$,
3. *transposition*: $(\forall x, y, z, w : A) m(m(x, y), m(z, w)) = m(m(x, z), m(y, w))$, and
4. *cancellation*: $(\forall x, y, z : A) m(x, z) = m(y, z) \Rightarrow x = y$.

The intuition is that A is an interval in a collection of real numbers, and that $m(x, y) = \frac{x+y}{2}$.

Next, Escardó–Simpson define a *right-iteration axiom* that may be understood as saying that m can be extended to the midpoint

$$M(x) = \frac{x_0}{2} + \frac{x_1}{4} + \frac{x_2}{8} \cdots = m(x_0, m(x_1, m(x_2, \ldots)))$$

of a sequence of points $x : A^{\mathbb{N}}$ in A. Escardó–Simpson state their axiom without reference to sequences or the natural numbers, by requiring that for every morphism $c : X \longrightarrow A \times X$, there exists a unique morphism $u : X \longrightarrow A$ making the following diagram commute.

$$\begin{tikzcd}
A \times X \arrow[r, "\mathrm{id} \times u"] & A \times A \arrow[d, "m"] \\
X \arrow[u, "c"] \arrow[r, "u"] & A
\end{tikzcd}$$

This definition involves quantifying over all objects in the category. To avoid impredicativity issues, we would like to phrase the type-theoretic analogue differently, which we can do using their Proposition 3.1, which gives an equivalent definition in the category of sets using sequences of elements of A, and this is a common definition in sufficiently expressive settings [39].

Definition 5.3.2. A *convex body* is a cancellative midpoint algebra (A, m) that satisfies the following *iteration property*: it comes equipped with a map $M : A^{\mathbb{N}} \longrightarrow A$ such that for every $x, y : A^{\mathbb{N}}$ we have

1. $M(x) = m(x_0, M(\lambda i.x_{i+1}))$, and
2. if $(\forall i : \mathbb{N}) y_i = m(x_i, y_{i+1})$ then $y_0 = M(x)$.

Lemma 5.3.3. *Given a cancellative midpoint algebra (A, m), the iteration property above is a proposition. By Lemma 2.6.20, this means that given two maps $M, N : A^{\mathbb{N}} \longrightarrow A$ satisfying the conditions above, we get $M = N$.*

Proof. Since we are showing the equality of functions, we use function extensionality, so that we may take $x : A^{\mathbb{N}}$. It suffices to show $M(x) = N(x)$.

We define the sequence $y : A^{\mathbb{N}}$ by

$$y_i :\equiv M(\lambda n.x_{n+i}).$$

Then we have $M(x) = y_0$. It remains to show that $y_0 = N(x)$, which we do using the second condition of Definition 5.3.2. So it suffices to prove $(\forall i : \mathbb{N}) y_i = m(x_i, y_{i+1})$. So let $i : \mathbb{N}$, and we compute:

$$\begin{aligned} y_i &= M(\lambda n.x_{n+i}) \\ &= m(x_i, M(\lambda n.x_{n+i+1})) \\ &= m(x_i, y_{i+1}). \end{aligned}$$ □

By the above lemma, the iteration property of Definition 5.3.2 is a proposition, justifying the usage of the word "property". Motivated by this result, we rephrase the iteration property more type-theoretically.

Theorem 5.3.4. *A cancellative midpoint algebra (A, m) satisfies the iteration property of Definition 5.3.2 iff the following proposition holds:*

$$(\Pi x : A^{\mathbb{N}}) \operatorname{isContr}((\Sigma \alpha : A^{\mathbb{N}})(\Pi i : \mathbb{N}) \alpha_i = m(x_i, \alpha_{i+1})).$$

The intuition is that while $M(x)$ represents the series of the entire sequence x, the element $\alpha : A^{\mathbb{N}}$ represents the series of all tails of the sequence, with $\alpha_0 = M(x)$.

Proof. Let (A, m) be a cancellative midpoint algebra with a map $M : A^{\mathbb{N}} \to A$ which satisfies the two conditions of Definition 5.3.2. We show that the stated proposition holds, so let $x : A^{\mathbb{N}}$.

To show that the type

$$(\Sigma \alpha : A^{\mathbb{N}})(\Pi i : \mathbb{N}) \alpha_i = m(x_i, \alpha_{i+1})$$

is a proposition, suppose we have $\alpha, \beta : A^{\mathbb{N}}$ with $(\Pi i : \mathbb{N}) \alpha_i = m(x_i, \alpha_{i+1})$ and $(\Pi i : \mathbb{N}) \beta_i = m(x_i, \beta_{i+1})$. It suffices to show $\alpha_k = \beta_k$ for $k : \mathbb{N}$ arbitrary. The sequences $y, y' : A^{\mathbb{N}}$ defined by

$$y_i = \alpha_{i+k} \qquad \text{and} \qquad y'_y = \beta_{i+k}$$

satisfy $(\forall i : \mathbb{N}) y_i = m(x_{i+k}, y_{i+1})$ and $(\forall i : \mathbb{N}) y'_i = m(x_{i+k}, y'_{i+1})$, so that by the second condition of Definition 5.3.2 we have $\alpha_k = y_0 = M(\lambda n.x_{n+k}) = y'_0 = \beta_k$.

To construct an element of

$$(\Sigma\alpha : A^{\mathbb{N}})(\Pi i : \mathbb{N})\alpha_i = m(x_i, \alpha_{i+1}),$$

set $\alpha_k := M(\lambda n.x_{n+k})$. Then $(\Pi i : \mathbb{N})\alpha_i = m(x_i, \alpha_{i+1})$ holds by the first condition of Definition 5.3.2, applied, for $i : \mathbb{N}$, to the sequence $\lambda n.x_{n+i}$.

Conversely, suppose that

$$f : (\Pi x : A^{\mathbb{N}})\ \mathrm{isContr}((\Sigma\alpha : A^{\mathbb{N}})(\Pi i : \mathbb{N})\alpha_i = m(x_i, \alpha_{i+1})),$$

with a pointwise center of contraction $f_c : (\Pi x : A^{\mathbb{N}})(\Sigma\alpha : A^{\mathbb{N}})(\Pi i : \mathbb{N})\alpha_i = m(x_i, \alpha_{i+1})$, and in particular a choice map $f_\alpha(x) := \mathrm{pr}_0(f_c(x))$ of type $A^{\mathbb{N}} \to A^{\mathbb{N}}$, and a map $f_p(x) := \mathrm{pr}_1(f_c(x))$ of type $(\Pi x : A^{\mathbb{N}})(\Pi i : \mathbb{N})f_\alpha(x)(i) = m(x_i, f_\alpha(x)(i+1))$. We set $M(x) := f_\alpha(x)(0)$.

To show the first condition for our definition of M, let $x : A^{\mathbb{N}}$. Since we have

$$f_p(x)(0) : f_\alpha(x)(0) = m(x_0, f_\alpha(x)(1)),$$

in order to show $M(x) = m(x_0, M(\lambda n.x_{n+1}))$, it suffices to show

$$f_\alpha(x)(1) = f_\alpha(\lambda n.x_{n+1})(0).$$

We show the more general property that the sequences

$$\alpha_i := f_\alpha(x)(i+1) \qquad \text{and} \qquad \beta_i := f_\alpha(\lambda n.x_{n+1})(i)$$

are equal. They satisfy $(\Pi i : \mathbb{N})\alpha_i = m(x_{i+1}, \alpha_{i+1})$ using $\lambda i.f_p(x)(i+1)$ and $(\Pi i : \mathbb{N})\beta_i = m(x_{i+1}, \beta_{i+1})$ using $f_p(\lambda n.x_{n+1})$. This yields two elements of the contractible type

$$(\Sigma\alpha : A^{\mathbb{N}})(\Pi i : \mathbb{N})\alpha_i = m(x_{i+1}, \alpha_{i+1}).$$

Because a contractible type is a proposition, we get $\alpha = \beta$ and in particular

$$M(x) = m(x_0, \alpha_0) = m(x_0, \beta_0) = m(x_0, M(\lambda n.x_{n+1})).$$

To show the second condition for our chosen M, let $x, y : A^{\mathbb{N}}$ with $(\forall i : \mathbb{N})y_i = m(x_i, y_{i+1})$, so that y is an element of the contractible type $(\Sigma\alpha : A^{\mathbb{N}})(\Pi i : \mathbb{N})\alpha_i = m(x_i, \alpha_{i+1})$. Hence, because a contractible type is a proposition, we get $y = f_\alpha(x)$ and in particular $y_0 = M(x)$. □

Theorem 5.3.4 restates the iteration property into a style commonly found in univalent type theory. This restatement is a proposition even if A is not a set. In the same spirit, it may be valuable to rephrase the cancellation property in Definition 5.3.1 into the requirement that for every $x : A$, the map $m(-, x) : A \to A$, or using symmetry equivalently the map $m(x, -) : A \to A$, is an embedding in the sense of Definition 2.6.2, so that the condition is a proposition even if A is not a set.

Such changes may result in definitions of convex bodies and interval objects that work for arbitrary underlying types A, rather than restricting to sets. However, we do not investigate this any further.

Definition 5.3.5. A *bipointed convex body* (A, m, a, b) is a convex body (A, m) with two points $a, b : A$. A homomorphism of convex bodies from (A, m, a, b) to (A', m', a', b') is a map $f : A \to A'$ on the underlying sets that preserves the structure in the sense that $f(a) = a'$, $f(b) = b'$, and $(\forall x, y : A)f(m(x, y)) = m'(f(x), f(y))$.

Finally, we are ready to define interval objects.

Definition 5.3.6. An *interval object* is a homotopy-initial bipointed convex body, i.e. a bipointed convex body (A, m, a, b) such that for any other convex body (A', m', a', b'), the type

$$(\Sigma f : A \to A')f(a) = a' \wedge f(b) = b' \wedge (\forall x, y : A)f(m(x, y)) = m'(f(x), f(y))$$

of homomorphisms from (A, m, a, b) to (A', m', a', x') is contractible.

> *Remark* 5.3.7. Since A' is a set, saying that the above type of homomorphisms is contractible is equivalent to saying that that there exists a homomorphisms, and any two homomorphisms are pointwise equal.

This concludes our rephrasing of the category-theoretic definition by Escardó–Simpson into type theory.

5.4 Notes

Thanks to our systematic use of Cauchy structures, and in particular Corollary 4.5.11, we have written a rather short new proof that the HoTT book reals coincide with the Euclidean reals in Proposition 5.1.3, without relying on propositional resizing, thus also improving on Booij [26].

In the presence of propositional resizing, we can define $\mathbb{R}_{\mathbf{E}}$. Theorem 5.2.1, showing that $\mathbb{R}_{\mathbf{E}}$ is a homotopy-initial Cauchy structure, without assuming that $\mathbb{R}_{\mathbf{H}}$ exists, is new. Two open questions remain in regard to this result:

1. Can we show homotopy-initiality with respect to arbitrary types equipped with Cauchy structures, rather than only sets? Note that a type with a Cauchy structure is not automatically a set: given a Cauchy structure on a type X, we can assign a Cauchy structure to $X + Y$ for an arbitrary type Y, with elements in the right disjunct being assigned an infinite distance to all elements.

2. What homotopy-initiality can be shown in the absence of propositional resizing, given only a least Cauchy complete subset of the Dedekind reals, without knowing its construction as an intersection of subsets of $\mathbb{R}_{\mathbf{D}}$?

The first type-theoretic definition of interval objects given by Definitions 5.3.1 and 5.3.2 is a naive translation of Escardó and Simpson [43, 39]. Type-theoretic language then allowed us to phrase uniqueness of the iteration operator M as a certain type being a proposition in Lemma 5.3.3, and we have subsequently given a new and equivalent definition of interval objects in Theorem 5.3.4.

The results of Sections 5.1 and 5.2 suggest that the interval in the HoTT book reals is an interval object, since in toposes the interval in the Euclidean reals is an interval object [43].

We obtain a convex body by following the intuition of the structure of convex bodies. Concretely, we take as the underlying set the interval $[-1, 1] \subseteq \mathbb{R}_{\mathbf{H}}$, i.e.

$$\mathbb{I}_{\mathbf{H}} :\equiv (\Sigma x : \mathbb{R}_{\mathbf{H}}) - 1 \leq x \leq 1,$$

with the usual identification of elements of $\mathbb{I}_\mathbf{H}$ with their underlying element of $\mathbb{R}_\mathbf{H}$, and $m(x, y) = \frac{x-y}{2}$, so that the convex body is given by $(\mathbb{I}_\mathbf{H}, m, -1, 1)$. It has to be checked that $(\mathbb{I}_\mathbf{H}, m, -1, 1)$ satisfies the property of a convex body.

Conjecture 5.4.1. $(\mathbb{I}_\mathbf{H}, m, -1, 1)$ *is an interval object.*

Conversely, we may also start with an arbitrary interval object $(\mathbb{I}, m, a, b)$. From this, we may define a candidate type $\mathbb{R}_\mathbf{I}$ of real numbers as a quotient of $\mathbb{Z} \times \mathbb{I}$, where we identify

1. $(k - 1, x)$ with (k, y) if $m(x, y) = m(m(a, b), a)$, and

2. (k, x) with $(k + 1, y)$ if $m(x, y) = m(m(a, b), b)$,

so that $(k, m(a, b))$ represents the real number k, and so that (k, a) is identified with $(k - 1, m(a, b))$ and similarly (k, b) with $(k + 1, m(a, b))$ and $(k + 2, a)$.

Conjecture 5.4.2. *Given an interval object $(\mathbb{I}, m, a, b)$, a Cauchy structure can be defined on the type $\mathbb{R}_\mathbf{I}$ constructed above, so that $\mathbb{R}_\mathbf{I}$ satisfies a homotopy-initiality property similar to the one for $\mathbb{R}_\mathbf{E}$ in Theorem 5.2.1. That is, for a Cauchy structure $(S, \sim)$, where S is a set, the type* $\text{CS-hom}(\mathbb{R}_\mathbf{I}, S)$ *of Cauchy structure morphisms from $\mathbb{R}_\mathbf{I}$ to S is contractible.*

If this works out, it may be possible to present the interval objects, and thus the HoTT book reals, as a higher-inductive type, rather than as a higher inductive-inductive type.

Chapter 6

LOCATORS

In this chapter we work with an arbitrary but fixed type of reals $\mathbb{R}$, namely any Cauchy complete Archimedean field in the sense of Chapter 4. A previous version of this work was developed for $\mathbb{R}_{\mathbf{D}}$ specifically [23].

The basic idea is that we equip real numbers with the structure of a *locator*, defined in Section 6.1. The purpose of the work is to show *how* to extract discrete information from an existing theory of real analysis in UTT.

The following example, which will be fully proved in Theorem 7.3.5, illustrates how we are going to use locators. Suppose f is a pointwise continuous function, and $a < b$ are real numbers with locators. Further suppose that f is locally nonconstant, that $f(x)$ has a locator whenever x has a locator, and that $f(a) \leq 0 \leq f(b)$. Then we can find a root of f, which comes equipped with a locator. For the moment, we provide a proof sketch, to motivate the techniques that we are going to develop in this section. We define sequences $a, b : \mathbb{N} \to \mathbb{R}$ with $a_n < a_{n+1} < b_{n+1} < b_n$, with $f(a_n) \leq 0 \leq f(b_n)$, with $b_n - a_n \leq (b-a)\left(\frac{2}{3}\right)^n$, and such that all a_n and b_n have locators. Set $a_0 = a$, $b_0 = b$. Suppose a_n and b_n are defined. We will explain in the complete proof of Theorem 7.3.5 how to to find q_n with $\frac{2a_n+b_n}{3} < q_n < \frac{a_n+2b_n}{3}$ and $f(q_n) \# 0$. The important point for the moment, is that this is possible precisely because we have locators.

- If $f(q_n) > 0$, then set $a_{n+1} :\equiv a_n$ and $b_{n+1} :\equiv q_n$.

- If $f(q_n) < 0$, then set $a_{n+1} :\equiv q_n$ and $b_{n+1} :\equiv b_n$.

The sequences converge to a number x. For any $\varepsilon : \mathbb{Q}_+$, we have $|f(x)| \leq \varepsilon$, hence $f(x) = 0$. This completes our sketch.

We need to explain why the sequences a and b come equipped with locators, and why their limit x has a locator. In fact, all q_n are rationals, and hence have locators, as discussed in Section 6.3. The number q_n is constructed using the central techniques for observing data from locators, see Sections 6.4 and 6.5. These techniques can then also be used in Section 6.6 to compute rational bounds. Locators for $\frac{2a_n+b_n}{3}$ and $\frac{a_n+2b_n}{3}$ can be constructed as locators for algebraic operations, as in Section 6.7. Locators for limits are discussed in Section 6.8.

We compute signed digit representations for reals with locators in Section 6.9. Given a real and a locator, we strengthen the properties for being a Dedekind cut into structure in Section 6.10.

6.1 Definition

Recall that the rationals are a subfield of $\mathbb{R}$ by Lemma 4.3.3. Throughout Chapters 6–8 we identify $q : \mathbb{Q}$ with its embedding $i(q) : \mathbb{R}$.

Recall from Definition 4.6.1 that a pair of predicates on the rationals $x = (L, U)$ is *located* if $(\forall q, r : \mathbb{Q})(q < r) \Rightarrow (q < x) \vee (x < r)$. Indeed, this property holds for an arbitrary $x : \mathbb{R}$ by cotransitivity of $<$.

Definition 6.1.1. A *locator* for $x : \mathbb{R}$ is a function $\ell : (\Pi q, r : \mathbb{Q}) q < r \to (q < x) + (x < r)$. We denote by $\mathrm{locator}(x)$ the type of locators on x. That is, we replace the logical disjunction in locatedness by a disjoint sum, so that we get structure rather than property, allowing us to compute.

Remark 6.1.2. It should be possible to substitute for $\mathbb{Q}$, throughout Chapters 6–9, the dyadic rationals, or other dense subsets of the reals satisfying suitable conditions, perhaps including approximate division as in Bauer and Taylor [13]. We use $\mathbb{Q}$, rather than

> a computationally more convenient type, in order to stay close to the traditional mathematical development.

A locator can be seen as an analogue to a Turing machine representing a computable real number, in the sense that it will provide us with enough data to be able to type-theoretically compute, for instance, signed-digit expansions. However, a locator does not express that a given real is a computable real: in the presence of excluded middle, there exists a locator for every $x : \mathbb{R}$, despite not every real being computable.

Lemma 6.1.3. *Assuming* PEM, *for every* $x : \mathbb{R}$, *we can construct a locator for* x.

Proof. For given rationals $q < r$, use PEM to decide $q < x$. If $q < x$ holds, we can simply return the proof given by our application of PEM. If $\neg(q < x)$ holds, then we get $x \leq q < r$ so that we can return a proof of $x < r$. □

> *Remark* 6.1.4. We recall from our discussion in Section 2.4.1 that we use the word "proof" also to refer to type-theoretic constructions of types that are not propositions. This chapter contains many such proofs that do not prove propositions in the sense of Definition 2.4.1.

In Chapter 7, we will define when a function $f : \mathbb{R} \to \mathbb{R}$ *lifts to locators*, which can be seen as an analogue to a computable function on the reals. There, the contrast with the theory of computable analysis becomes more pronounced, as the notion of lifting to locators is neither stronger nor weaker than continuity.

The structure of a locator has been used previously by The Univalent Foundations Program in a proof that assuming either countable choice or excluded middle, the Cauchy reals and the Dedekind reals coincide [91, Section 11.4].

The reader may wonder why we only choose to modify one of the Dedekind properties to become structure. We show in Theorem 6.10.3 that given only a locator, we can obtain the remaining structures, corresponding to boundedness, roundedness and transitivity, automatically.

6.2 Terminology for locators

A locator ℓ for a real x can be evaluated by picking $q, r : \mathbb{Q}$ and $\nu : q < r$. The value $\ell(q, r, \nu)$ has type $(q < x) + (x < r)$, and so $\ell(q, r, \nu)$ can be either in the left summand or the right summand. We say that "we locate $q < x$" when the locator gives a value in the left summand, and similarly we say "we locate $x < r$" when the locator gives a value in the right summand.

We often do case analysis on $\ell(q, r, \nu) : (q < x) + (x < r)$ by constructing a value $c : C(q <_x r)$ for some type family $C : (q < x) + (x < r) \to \mathcal{U}$. To construct c we use the elimination principle of +, for which we need to specify two values corresponding to the disjuncts $q < x$ and $x < r$, so the two values have corresponding types $(\Pi \xi : q < x) C(\mathsf{inl}(\xi))$ and $(\Pi \zeta : x < r) C(\mathsf{inr}(\zeta))$. These two values correspond to the two possible answers of the locator, and we will often indicate this by using the above terminology: the expression "we locate $q < x$" corresponds to constructing a value of the former type, and the expression "we locate $x < r$" corresponds to constructing a value of the latter type.

For example, for every real x with a locator ℓ, we can output a Boolean depending on whether ℓ locates $0 < x$ or $x < 1$. Namely, if we locate $0 < x$ we output true, and if we locate $x < 1$ we output false. We use this construction in the proof of Lemma 6.10.1.

6.3 Locators for rationals

Lemma 6.3.1. *Suppose $x : \mathbb{R}$ is a rational, or more precisely, that $(\exists s : \mathbb{Q})(x = i(s))$, with $i : \mathbb{Q} \hookrightarrow \mathbb{R}$ the canonical inclusion obtained from Lemma 4.3.3, then x has a locator.*

We give two constructions, to emphasize that locators are not unique. We use trichotomy of the rationals, namely, for all $a, b : \mathbb{Q}$,

$$(a < b) + (a = b) + (a > b).$$

First proof. Let $q < r$ be arbitrary, then we want to give $(q < s) + (s < r)$. By trichotomy of the rationals applied to q and s, we have

$$(q < s) + (q = s) + (q > s)$$

In the first case $q < s$, we can locate $q < s$. In the second case $q = s$, we have $s = q < r$, so we locate $s < r$. In the third case, we have $s < q < r$, so we locate $s < r$. □

Second proof. Let $q < r$ be arbitrary, then we want to give $(q < s) + (s < r)$. By trichotomy of the rationals applied to s and r, we have

$$(s < r) + (s = r) + (s > r)$$

In the first case $s < r$, we can locate $s < r$. In the second case $s = r$, we have $q < r = s$, so we locate $q < s$. In the third case, we have $q < r = s$, so we locate $q < s$. □

In the case that $q < s < r$, the first construction locates $s < r$, whereas the second construction locates $q < s$. In particular, given a pair $q < r$ of rationals, the first proof locates $q < 0$ if q is indeed negative, and $0 < r$ otherwise. The second proof locates $0 < r$ if r is indeed positive, and $q < 0$ otherwise. Note that these locators disagree when $q < 0 < r$, illustrating that locators are not unique.

6.4 The logic of locators

Our aim is to combine properties of real numbers with the structure of a locator to make discrete observations.

If one *represents* reals by Cauchy sequences, one obtains lower bounds immediately from the fact that any element in the sequence approximates the real up to a known error. As a working example, we show, perhaps surprisingly, that we can get a lower bound for an real x, that is an element of $(\Sigma q : \mathbb{Q})q < x$, from the locator alone.

Recall that Dedekind reals are bounded from below, so that $(\exists q : \mathbb{Q})q < x$. For an arbitrary $x : \mathbb{R}$, we know that $x - 1 < x$, and hence by the Archimedean property we have $(\exists q : \mathbb{Q})x - 1 < q < x$, and so we get lower boundedness $(\exists q : \mathbb{Q})q < x$ as a consequence of this. We will define a proposition P which *gives* us a bound, in the sense that we can use the elimination rule for propositional truncations to get a map

$$((\exists q : \mathbb{Q}) q < x) \to P,$$

and then we can extract a bound using a simple projection map

$$P \to ((\Sigma q : \mathbb{Q}) q < x).$$

More concretely, we define a type of rationals which are bounds for x and which are *minimal* in a certain sense. The minimality is *not* intended to find tight bounds, but is intended to make this collection of rationals into a proposition: in other words, minimality ensures that the answer is unique, so that we may apply the elimination rule for propositional truncations.

Our technique has two central elements: reasoning about the structure of locators using propositions, and the construction of a unique answer using bounded search (Section 6.5).

Given a locator $\ell : \mathrm{locator}(x)$, $q, r : \mathbb{Q}$ and $v : q < r$, we have the notation

$$q <^{\ell}_{x} r :\equiv \ell(q, r, v) : (q < x) + (x < r),$$

leaving the proof of $q < r$ implicit. We further often drop the choice of locator, writing $q <_x r$ for $q <^{\ell}_{x} r$.

Recall the type $\mathrm{DHProp} :\equiv (\Sigma P : \mathrm{HProp}) P + \neg P$ of decidable propositions from Definition 2.4.9.

Lemma 6.4.1. *For types A and B, we have*

$$A + B \simeq (\Sigma P : \mathrm{DHProp})(P \to A) \times (\neg P \to B).$$

Proof. For a given element $x : A + B$, the proposition P is defined to hold when x an given by an element of A, and false otherwise, so that the two conditions on P hold. Vice versa, for a given proposition P we simply decide P to obtain the respective element of $A + B$. It has to be checked that these two constructions result in an equivalence. □

Lemma 6.4.2. *The type* $\mathrm{locator}(x)$ *of Definition 6.1.1 is equivalent to the type*

$$\begin{aligned}&(\Sigma \mathrm{locatesRight} : (\Pi q, r : \mathbb{Q}) q < r \to \mathrm{DHProp})\\&\quad((\Pi q, r : \mathbb{Q})(\Pi v : q < r)\, \mathrm{locatesRight}(q, r, v) \to q < x)\\&\times ((\Pi q, r : \mathbb{Q})(\Pi v : q < r) \neg\, \mathrm{locatesRight}(q, r, v) \to x < r).\end{aligned}$$

Proof. The previous lemma yields the equivalence

$$\begin{aligned}\mathrm{locator}(x) \simeq (\Pi q, r : \mathbb{Q}) q < r \to&\\&(\Sigma P : \mathrm{DHProp})(P \to q < x) \times (\neg P \to x < r),\end{aligned}$$

and then we can apply Theorems 2.15.5 and 2.15.7 in The Univalent Foundations Program [91] to distribute the Π-types over Σ and ×. □

Remark 6.4.3. We emphasize that, confusingly, $\mathrm{locatesRight}(q, r, v)$ is defined type-theoretically as $\mathrm{isLeft}(q <^{\ell}_{x} r)$.

Definition 6.4.4. For a real x with a locator ℓ and rationals $q < r$, we write

$$\mathrm{locatesRight}(q <^{\ell}_{x} r) \qquad \text{or} \qquad \mathrm{locatesRight}(q <_{x} r)$$

for the decidable proposition $\mathrm{locatesRight}(q, r, v)$ obtained from Lemma 6.4.2. We write

$$\mathrm{locatesLeft}(q <^{\ell}_{x} r) \qquad \text{or} \qquad \mathrm{locatesLeft}(q <_{x} r)$$

to be the negation of $\mathrm{locatesRight}(q <_{x} r)$: so it is the proposition which is true if we locate $x < r$.

Remark 6.4.5. In general, if we have $q' < q < r$, then $\mathrm{locatesRight}(q <_{x} r)$ does *not* imply $\mathrm{locatesRight}(q' <_{x} r)$.

Lemma 6.4.6. *For any real x with a locator ℓ and rationals $q < r$,*

$$\neg(q < x) \Rightarrow \text{locatesLeft}(q <^{\ell}_{x} r), \qquad \textit{and}$$
$$\neg(x < r) \Rightarrow \text{locatesRight}(q <^{\ell}_{x} r).$$

Proof. From the defining properties of locatesRight in Lemma 6.4.2, we know

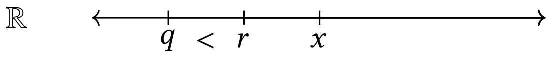
$$\text{locatesRight}(q <^{\ell}_{x} r) \Rightarrow (q < x), \qquad \text{and}$$
$$\neg\,\text{locatesRight}(q <^{\ell}_{x} r) \Rightarrow (x < r).$$

The contrapositives of these are, respectively:

$$\neg(q < x) \Rightarrow \neg\,\text{locatesRight}(q <^{\ell}_{x} r), \qquad \text{and}$$
$$\neg(x < r) \Rightarrow \neg\neg\,\text{locatesRight}(q <^{\ell}_{x} r).$$

Using the fact that $\neg\neg A \Rightarrow A$ when A is decidable, this is the required result. □

Example 6.4.7. Let x be a real equipped with a locator. We can type-theoretically express that the locator must give certain answers. For example, if we have $q < r < x$, shown visually as



we must locate $q < x$, because $\neg(x < r)$. In other words, we obtain truth of the proposition locatesRight($q <_x r$): the property $\neg(x < r)$ yielded a property of the structure $q <_x r$.

Continuing our working example of computing a lower bound, for any $q : \mathbb{Q}$ we have the claim

$$P(q) :\equiv \text{locatesRight}(q - 1 <_x q)$$

that we locate $q - 1 < x$. This claim is a decidable proposition. And from the existence $(\exists q : \mathbb{Q}) q < x$ of a lower bound for x, we can deduce that $(\exists q : \mathbb{Q}) P(q)$, because if $q < x$ then $\neg(x < q)$ and hence the above lemma applies. If we manage to *find* a $q : \mathbb{Q}$ for which $P(q)$ holds, then we have certainly found a lower bound of x, namely $q - 1$.

6.5 Bounded search

Even though the elimination rule for propositional truncation in Definition 2.4.3 only constructs maps into propositions, we can use elements of propositional truncations to obtain witnesses of non-truncated types — in other words, we can sometimes obtain structure from property.

Theorem 6.5.1 (Escardó [40], [44], [91, Exercise 3.19])**.** *Let $P : \mathbb{N} \to \mathrm{DHProp}$. If $(\exists n : \mathbb{N})P(n)$ then we can construct an element of $(\Sigma n : \mathbb{N})P(n)$.*

Proof. Define the type of *least* numbers satisfying P as $(\Sigma n : \mathbb{N})P(n) \times \mathrm{minimal}(n, P)$, where

$$\mathrm{minimal}(n, P) \coloneqq (\forall k : \mathbb{N})k \leq n \Rightarrow P(k) \Rightarrow n = k$$

expresses that n is minimal with respect to P, and observe that $(\Sigma n : \mathbb{N})P(n) \times \mathrm{minimal}(n, P)$ is a proposition. We have

$$((\Sigma n : \mathbb{N})P(n)) \to ((\Sigma n : \mathbb{N})P(n) \times \mathrm{minimal}(n, P))$$

by a bounded search: given a natural number n that satisfies P, we can find the *least* natural number satisfying P, by searching up to n. Using the elimination rule for propositional truncations, we obtain the dashed vertical map in the following diagram.

$$\begin{tikzcd}
(\Sigma n : \mathbb{N})P(n) \arrow[r, "|\cdot|"] \arrow[rd, "\text{bounded search}"'] & (\exists n : \mathbb{N})P(n) \arrow[d, dashed, "\exists!"] \\
& (\Sigma n : \mathbb{N})P(n) \times \mathrm{minimal}(n, P) \arrow[d, "\mathrm{proj}"] \\
& (\Sigma n : \mathbb{N})P(n)
\end{tikzcd}$$

The vertical composition is the required result. □

Remark 6.5.2. There are different ways to obtain an element of $(\Sigma n : \mathbb{N})P(n)$ from an element of $(\exists n : \mathbb{N})P(n)$.

1. The output doesn't have to be the smallest natural satisfying P, since you could

choose a strange ordering on the naturals, and, for example, search backwards between 0 and 100, and ordinary (increasing) above 100.

2. The output doesn't even have to be the minimal number — not even in a strange ordering of the naturals. For a (contrived) example, an ill-informed search could choose to always test $P(0)$ through $P(5)$, and output the least i between 0 and 4 for which $P(i)$ and $P(i+1)$ are both true, if it exists, and otherwise do ordinary bounded search starting from 0. This computation always succeeds since, in the worst case, we fall back to the bounded search, which we know works. But the output value is not minimal in any sense, since it may output 4 even if $P(0)$ also holds (but $P(1)$ doesn't), but may also output 0 even if $P(4)$ also holds (but $P(5)$ doesn't).

Remark 6.5.3. In general, we don't have $\|X\| \to X$ for all types X, as this would imply excluded middle [63]. But for some types X, we do have $\|X\| \to X$, namely when X has a constant endomap [63].

Even without univalence, Theorem 6.5.1 also works for any type equivalent to $\mathbb{N}$.

Corollary 6.5.4. *Let A be a type and $e : \mathbb{N} \simeq A$ be an equivalence, that is, a function $\mathbb{N} \to A$ with a left inverse and right inverse. Let $P : A \to \mathrm{DHProp}$. If $(\exists a : A)P(a)$ then we can construct an element of $(\Sigma a : A)P(n)$.*

Proof. Use Theorem 6.5.1 with $P'(n) :\equiv P(e(n))$. In order to show $(\exists n : \mathbb{N})P'(n)$, it suffices to show $((\Sigma a : A)P(a)) \to ((\Sigma n : \mathbb{N})P'(n))$, so let $a : A$ and $p : P(a)$. Then since $a = e(e^{-1}(a))$ we get $P(e(e^{-1}(a)))$ by transport.

Hence from Theorem 6.5.1 we obtain some $(n, p') : (\Sigma n : \mathbb{N})P'(e(n))$, so we can output $(e(n), q)$. □

6.6 Computing bounds

We are now ready to finish our running example of computing a lower bound for x.

Lemma 6.6.1. *Given a real $x : \mathbb{R}$ equipped with a locator, we get bounds for x, that is, we can find $q, r : \mathbb{Q}$ with $q < x < r$.*

Proof. We pick any enumeration of $\mathbb{Q}$, that is, an equivalence $\mathbb{N} \simeq \mathbb{Q}$. Set

$$P(q) :\equiv \mathrm{locatesRight}(q - 1 <_x q).$$

From Section 6.4 we know that $(\exists q : \mathbb{Q})P(q)$, and so we can apply Corollary 6.5.4. We obtain $(\Sigma q : \mathbb{Q})P(q)$, and in particular $(\Sigma q : \mathbb{Q})q - 1 < x$.

Upper bounds are constructed by a symmetric argument, using

$$P(r) :\equiv \mathrm{locatesLeft}(r <_x r + 1). \qquad \square$$

We emphasize that even though we cannot decide $q < x$ in general, we *can* decide what the locator tells us, and this is what is exploited in our development. Given a real x with a locator, the above construction of a lower bound searches for a rational q for which we locate $q - 1 < x$. We emphasize once more that the rational thus found is minimal in the sense that it appears first in the chosen enumeration of $\mathbb{Q}$, and *not* a tight bound.

Remark 6.6.2. The proof of Theorem 6.5.1 works by an exhaustive, but bounded, search. So our construction for Lemma 6.6.1 similarly exhaustively searches for an appropriate rational q. The efficiency of the algorithm thus obtained can be improved:

1. We do not need to test every rational number: it suffices to test, for example, bounds of the form $\pm 2^{k+1}$ for $k : \mathbb{N}$, as there always exists a bound of that form. Formally, such a construction is set up by enumerating a subset of the integers instead of enumerating all rationals, and showing the existence of a bound of the chosen form, followed by application of Corollary 6.5.4.

2. More practically, Lemma 6.6.1 shows that we may as well additionally equip bounds to reals that already have locators. Then, any later constructions that use rational bounds can simply use these equipped rational bounds. This is essentially the approach of interval arithmetic with open nondegenerate intervals. We can also see

this equipping of bounds as a form of memoization, which we can apply more generally.

Lemma 6.6.3. *For a real x equipped with a locator and any positive rational ε we can find $u, v : \mathbb{Q}$ with $u < x < v$ and $v - u < \varepsilon$.*

Proof. The construction of bounds in Lemma 6.6.1 yields $q, r : \mathbb{Q}$ with $q < x < r$. We can compute $n : \mathbb{N}$ such that $r < q + \frac{n\varepsilon}{3}$. Consider the equidistant subdivision

$$q - \frac{\varepsilon}{3},\ q,\ q + \frac{\varepsilon}{3},\ q + \frac{2\varepsilon}{3},\ \ldots,\ q + \frac{n\varepsilon}{3},\ q + \frac{(n+1)\varepsilon}{3}.$$

By Lemma 6.4.6, necessarily $\mathrm{locatesRight}(q - \frac{\varepsilon}{3} <_x q)$ because $q < x$. Similarly, we have $\mathrm{locatesLeft}\,(q + \frac{n\varepsilon}{3} <_x q + \frac{(n+1)\varepsilon}{3})$ because $x < q + \frac{n\varepsilon}{3}$.

For some i, which we can find by a finite search using a discrete intermediate value theorem such as the one-dimensional version of Sperner's lemma, we have

$$\mathrm{locatesRight}\left(q + \frac{i\varepsilon}{3} <_x q + \frac{(i+1)\varepsilon}{3}\right) \land \mathrm{locatesLeft}\left(q + \frac{(i+1)\varepsilon}{3} <_x q + \frac{(i+2)\varepsilon}{3}\right).$$

For this i, we can output $u = q + \frac{i\varepsilon}{3}$ and $v = q + \frac{(i+2)\varepsilon}{3}$. □

Remark 6.6.4. The above result allows us to compute arbitrarily precise bounds for a real number x with a locator. But, as in Remark 6.6.2, the above theorem shows that we may as well *equip* an appropriate algorithm for computing arbitrarily precise lower and upper bounds to real numbers. This may be a better idea when efficiency of the computation matters. We come back to this in Remark 6.10.4.

6.7 Locators for algebraic operations

If x and y are reals that we can compute with in an appropriate sense, then we expect to be able to do so with $-x$, $x + y$, $x \cdot y$, x^{-1} (assuming $x \# 0$), $\min(x, y)$ and $\max(x, y)$ as well. In our case, that means that if x and y come equipped with locators, then so should the previously listed values.

If one works with intensional real numbers, such as when they are given as Cauchy sequences, then the algebraic operations are specified directly on the representations. This means that the computational data is automatically present. Since in our case the algebraic operations are specified extensionally, they do not give any discrete data, and so the construction of locators has to be done explicitly in order to compute.

If these algebraic operations are *defined* as specific values, such as when $x + y$ is defined as a pointwise sum of two Cauchy sequences, or as a certain Dedekind cuts combining those of x and y, then it is clear what work needs to be done to construct a locator. However, since $\mathbb{R}$ and the operations on it are arbitrary, we first need to characterize the algebraic operations in terms of the strict order $<$.

Recall from Section 4.6 that for a Dedekind cut $x = (L, U)$ we write $q < x$ for the claim that $q : \mathbb{Q}$ is in the left cut L. In fact, now that we have identified $q : \mathbb{Q}$ with its canonical embedding $i(q) : \mathbb{R}$ in the reals, we can simply understand $q < x$ as $i(q) <_{\mathbb{R}} x$, which coincides with the notation for Dedekind cuts.

The algebraic operations can be defined for Dedekind cuts as in The Univalent Foundations Program [91, Section 11.2.1]. Those same *definitions*, when written in our notation, become *properties* that we can prove for our arbitrary notion of reals. This was previously observed in The Univalent Foundations Program [91, Theorem 11.2.14]. Considering that we start with an arbitrary notion of real numbers, it is surprising that, at least in terms of the strict order, the algebraic operations necessarily coincide with the defined operations on the Dedekind reals.

Recall that $\mathbb{R}$ is Archimedean, which can be succinctly stated as the claim that for all $x, y : \mathbb{R}$,

$$x < y \Rightarrow (\exists q : \mathbb{Q}) x < q < y.$$

Lemma 6.7.1. *For $x, y, z, w : \mathbb{R}$ with $w < 0 < z$ and $q, r : \mathbb{Q}$ we have:*

$$q < -x \Leftrightarrow x < -q$$

$$-x < r \Leftrightarrow -r < x$$

$$q < x + y \Leftrightarrow (\exists s : \mathbb{Q}) s < x \wedge (q - s) < y$$

$$x + y < r \Leftrightarrow (\exists t : \mathbb{Q}) x < t \wedge y < (r - t)$$

$$q < xy \Leftrightarrow (\exists a, b, c, d : \mathbb{Q}) q < \min(ac, ad, bc, bd)$$

$$
\begin{aligned}
&\wedge a < x < b \wedge c < y < d \\
xy < r &\Leftrightarrow (\exists a, b, c, d : \mathbb{Q}) \max(ac, ad, bc, bd) < r \\
&\wedge a < x < b \wedge c < y < d \\
q < z^{-1} &\Leftrightarrow qz < 1 \\
z^{-1} < r &\Leftrightarrow 1 < rz \\
q < w^{-1} &\Leftrightarrow 1 < qw \\
w^{-1} < r &\Leftrightarrow rw < 1 \\
q < \min(x, y) &\Leftrightarrow q < x \wedge q < y \\
\min(x, y) < r &\Leftrightarrow x < r \vee y < r \\
q < \max(x, y) &\Leftrightarrow q < x \vee q < y \\
\max(x, y) < r &\Leftrightarrow x < r \wedge y < r
\end{aligned}
$$

Proof. $q < -x \Leftrightarrow x < -q$ holds by compatibility of + with <.

Suppose $q < x + y$. We aim to show $(\exists s : \mathbb{Q}) s < x \wedge (q - s) < y$. Using the Archimedean property we get $(\exists \varepsilon : \mathbb{Q}_{+}) q + \varepsilon < x + y$. Again by the Archimedean property $(\exists s : \mathbb{Q}) s < x < s + \varepsilon$, and in particular $-s < -x + \varepsilon$. Then we have $q + \varepsilon - s < x + y - x + \varepsilon = y + \varepsilon$, hence $q - s < y$, as required.

Conversely, suppose $(\exists s : \mathbb{Q}) s < x \wedge (q - s) < y$, then $q < x + y$ holds by compatibility of + with <.

For multiplication, suppose $q < xy$. As in the case for +, we first note that $(\exists \varepsilon : \mathbb{Q}_{+}) q + \varepsilon < xy$. Then note that $(\exists z : \mathbb{Q}_{+}) |x| + 1 < z \wedge |y| + 1 < z$. Set $\delta := \min(1, \frac{\varepsilon}{2z})$. Now $(\exists a, b, c, d : \mathbb{Q}) a < x < b \wedge c < y < d \wedge b - a < \delta \wedge d - c < \delta$ using a number of applications of the Archimedean property. It suffices to show that $q < \min(ac, ad, bc, bd)$, which can be done by showing it for each of the four cases. For instance, to show that $q < ac$, since $q + \varepsilon < xy$, it suffices to show $xy - \varepsilon < ac$, that is, that $xy < xy - x(y - c) - (x - a)y + (x - a)(y - c) + \varepsilon$, i.e. that $x(y - c) + (x - a)y < (x - a)(y - c) + \varepsilon$. But this holds since $x(y - c) + (x - a)y < |x| \delta + |y| \delta < \varepsilon$.

Conversely, suppose $(\exists a, b, c, d : \mathbb{Q}) q < \min(ac, ad, bc, bd) \wedge a < x < b \wedge c < y < d$. Since a, b, c, d are rationals, the minimum of the products is equal to one of the four products. For example, if $\min(ac, ad, bc, bd) = ac$, then $ac \leq ad$ gives $0 \leq a$, and $ac \leq bc$ gives $0 \leq c$. Hence $q < ac \leq ay < xy$. The remaining three cases are similar.

The laws for multiplicative inverses hold by compatibility of multiplication with $<$.

Suppose $q < \min(x, y)$. Without loss of generality we show $q < x$. By cotransitivity of $<$ we have $q < x \vee x < \min(x, y)$. The former case is the desired conclusion. For the latter case, recall from the definition of lattice that we have

$$\min(x, y) \leq \min(x, y) \Leftrightarrow \min(x, y) \leq x \wedge \min(x, y) \leq y,$$

and so since the left-hand side is tautologically true, we get $\min(x, y) \leq x$, contradicting $x < \min(x, y)$.

Conversely, suppose $q < x \wedge q < y$. Using cotransitivity twice we get $q < \min(x, y) \vee \min(x, y) < x$ and $q < \min(x, y) \vee \min(x, y) < y$. In either former case we are done, so the remaining case is that we have both $\min(x, y) < x$ and $\min(x, y) < y$. We know that

$$x \leq \min(x, y) \Leftrightarrow x \leq x \wedge x \leq y,$$

$$y \leq \min(x, y) \Leftrightarrow y \leq x \wedge y \leq y.$$

By the fact that $\min(x, y) < x$ and $\min(x, y) < y$, the left hand sides of these claims are false, and hence we have $\neg(x \leq y)$ and $\neg(y \leq x)$, which together contradict that $<$ is transitive and irreflexive.

Next we show from $\min(x, y) < r$ that $x < r \vee y < r$, so assume $\min(x, y) < r$. By cotransitivity we have $\min(x, y) < x$ or $x < r$, and similarly we have $\min(x, y) < y$ or $y < r$. Now in either latter case we are done, so we may assume to have both $\min(x, y) < x$ and $\min(x, y) < y$. By the same reasoning as before, this is a contradiction.

Conversely, suppose without loss of generality, that we have $x < r$. By cotransitivity we have $x < \min(x, y)$ or $\min(x, y) < r$. In the latter case we are done, and the former case contradicts

$$\min(x, y) \leq \min(x, y) \Leftrightarrow \min(x, y) \leq x \wedge \min(x, y) \leq y,$$

as above.

The laws not shown explicitly can be shown in a symmetric fashion. □

We will use the following variation of the Archimedean property.

Lemma 6.7.2. *For real numbers $x < y$, there* exist *$q : \mathbb{Q}$ and $\varepsilon : \mathbb{Q}_+$ with $x < q - \varepsilon < q + \varepsilon < y$.*

Proof. By a first application of the Archimedean property, we know $(\exists s : \mathbb{Q}) x < s < y$. Since we are showing a proposition, we may assume to have such an $s : \mathbb{Q}$. Now for $s < y$, by the Archimedean property, we know $(\exists t : \mathbb{Q}) s < t < y$, and again we may assume to have such a t. Now set $q :\equiv \frac{s+t}{2}$ and $\varepsilon :\equiv \frac{t-s}{2}$. □

In particular, the above variation can be used to strengthen the $\exists$ of the Archimedean property into Σ when the reals involved come equipped with locators. Its corollary, Corollary 6.7.4, is used to compute locators for multiplicative inverses.

Lemma 6.7.3. *For reals x and y equipped with locators we have the Archimedean* structure

$$x < y \to (\Sigma q : \mathbb{Q}) x < q < y.$$

Proof. Let x and y be reals equipped with locators. By Lemma 6.7.2, there exist $q : \mathbb{Q}$ and $\varepsilon : \mathbb{Q}_+$ with $x < q - \varepsilon < q + \varepsilon < y$. The following proposition is decidable for any (q', ε') and we have $(\exists (q, \varepsilon) : \mathbb{Q} \times \mathbb{Q}_+) P(q, \varepsilon)$:

$$P(q', \varepsilon') :\equiv \mathrm{locatesLeft}(q' - \varepsilon' <_x q') \wedge \mathrm{locatesRight}(q' <_y q' + \varepsilon').$$

Using Corollary 6.5.4 we can find (q', ε') with $P(q', \varepsilon')$ and hence $x < q' < y$. □

Corollary 6.7.4. *For reals x and y equipped with locators, and $s : \mathbb{Q}$ a rational, if $x < y$ then we have a choice of $x < s$ or $s < y$, that is:*

$$(\Pi s : \mathbb{Q}) x < y \to (x < s) + (s < y).$$

Proof. By Lemma 6.7.3 we can find $q : \mathbb{Q}$ with $x < q < y$. Apply trichotomy of the rationals: if $q < s$ or $q = s$ then we locate $x < s$, and if $s < q$ then we locate $s < y$. □

Remark 6.7.5. Instead of the rational $s : \mathbb{Q}$ we can have any real z equipped with a locator in the above corollary, so that we obtain a form of strong cotransitivity of the strict ordering relation on the real numbers, but we will not be using this.

Having developed such a strong cotransitivity, the proof of Lemma 6.7.1 could be carried out using the Archimedean *structure* of Lemma 6.7.3 rather than using the Archimedean property. This would then yield a structural characterization of the algebraic operations for $x, y : \mathbb{R}$ equipped with locators, along the lines of:

$$\begin{aligned}
q < x + y &\quad\Leftrightarrow\quad (\Sigma s, t : \mathbb{Q})(q = s + t) \wedge s < x \wedge t < y \\
q < x \cdot y &\quad\Leftrightarrow\quad (\Sigma a, b, c, d : \mathbb{Q}) q < \min(a \cdot c, a \cdot d, b \cdot c, b \cdot d) \\
&\qquad\qquad \wedge a < x < b \wedge c < y < d \\
q < \max(x, y) &\quad\Leftrightarrow\quad q < x + q < y \\
&\quad\vdots
\end{aligned}$$

Theorem 6.7.6. *If reals $x, y : \mathbb{R}$ are equipped with locators, then we can also equip $-x$, $x + y$, $x \cdot y$, x^{-1} (assuming $x \# 0$), $\min(x, y)$ and $\max(x, y)$ with locators.*

Remark 6.7.7. As we define absolute values by $|x| = \max(x, -x)$, as is common in constructive analysis, if x has a locator, then so does $|x|$, and we use this fact in the proof of the above theorem.

Proof of Theorem 6.7.6. Throughout this proof, we assume x and y to be reals equipped with locators, and $q < r$ to be rationals.

We construct a locator for $-x$. We can give $(q < -x) + (-x < r)$ by considering $-r <_x -q$.

We construct a locator for $x + y$. We need to show $(q < x + y) + (x + y < r)$. Note that $q < x + y$ iff there exists $s : \mathbb{Q}$ with $q - s < x$ and $s < y$. Similarly, $x + y < r$ iff there exists $t : \mathbb{Q}$ with $x < r - t$ and $y < t$.

Set $\varepsilon :\equiv (r - q)/2$, such that $q + \varepsilon = r - \varepsilon$. By Lemma 6.6.3 we can find $u, v : \mathbb{Q}$ such that $u < x < v$ and $v - u < \varepsilon$, so in particular $x < u + \varepsilon$. Set $s :\equiv q - u$, so that $q - s < x$. Now consider $s <_y s + \varepsilon$. If we locate $s < y$, we locate $q < x + y$. If we locate $y < s + \varepsilon$, we have $x < q - s + \varepsilon = r - s - \varepsilon$, that is, we can set $t :\equiv s + \varepsilon$ to locate $x + y < r$.

We construct a locator for $\min(x, y)$. We consider both $q <_x r$ and $q <_y r$. If we locate $x < r$ or $y < r$, we can locate $\min(x, y) < r$. Otherwise, we have located both $q < x$ and $q < y$, so we can locate $q < \min(x, y)$.

The locator for $\max(x, y)$ is symmetric to the case of $\min(x, y)$.

We construct a locator for xy. We need to show $(q < xy) + (xy < r)$. Note that $q < xy$ means:

$$(\exists a, b, c, d : \mathbb{Q})(a < x < b) \wedge (c < y < d) \wedge (q < \min\{ac, ad, bc, bd\}).$$

Similarly, $xy < r$ means:

$$(\exists a, b, c, d : \mathbb{Q})(a < x < b) \wedge (c < y < d) \wedge (\max\{ac, ad, bc, bd\} < r).$$

Using Lemma 6.6.3 we can find $z, w : \mathbb{Q}$ with $|x| + 1 < z$ and $|y| + 1 < w$, since we have already constructed locators for max, +, − and all rationals.

Set $\varepsilon :\equiv r - q$, $\delta :\equiv \min\{1, \frac{\varepsilon}{2z}\}$ and $\eta :\equiv \min\{1, \frac{\varepsilon}{2w}\}$. Find $a < x < b$ and $c < y < d$ such that $b - a < \eta$ and $d - c < \delta$. Note that $|a| < |x| + \eta \leq |x| + 1 < z$ and similarly $|b| < z$, $|c| < w$ and $|d| < w$. Then the distance between any two elements of $\{ac, ad, bc, bd\}$ is less than ε. For instance, $|ac-bd| < \varepsilon$ because $|ac-bd| \leq |ac-ad|+|ad-bd|$, and $|ac-ad| = |a||c-d| < |a|\delta < \frac{\varepsilon}{2}$ and similarly $|ad - bd| < \frac{\varepsilon}{2}$. Hence $\max\{ac, ad, bc, bd\} - \min\{ac, ad, bc, bd\} < \varepsilon$. Thus, by dichotomy of the rationals, one of $q < \min\{ac, ad, bc, bd\}$ and $\max\{ac, ad, bc, bd\} < r$ must be true, yielding a corresponding choice of $(q < xy) + (xy < r)$.

We construct a locator for x^{-1}. Consider the case that $x > 0$. Given $q < r$, we need $(q < x^{-1}) + (x^{-1} < r)$, or equivalently $(qx < 1) + (1 < rx)$. By the previous case, qx and rx have locators, so we can apply Corollary 6.7.4. The case $x < 0$ is similar. □

Remark 6.7.8. Locators for reciprocals can also be constructed by more elementary methods, as follows. For $x > 0$, we use dichotomy of the rationals for 0 and q. If $q \leq 0$ we may locate $q < x$, and otherwise we have $0 < 1/r < 1/q$, so that by considering $1/r <_x 1/q$ we may either locate $x < r$ or $q < x$. There is a similar construction for $x < 0$.

By deducing the same properties for the algebraic operations as the definitions in The Univalent Foundations Program [91, Section 11.2.1], and using the techniques of Sections 6.4 and 6.5, we have computed locators for algebraic operations applied to reals equipped with locators.

6.8 Locators for limits

In a spirit similar to the previous section, if we have a Cauchy sequence of reals, each of which equipped with a locator, then we can compute a locator for the limit of the sequence.

Lemma 6.8.1. *Suppose $x : \mathbb{N} \to \mathbb{R}$ has modulus of Cauchy convergence M, and suppose that every value in the sequence $x : \mathbb{N} \to \mathbb{R}$ comes equipped with a locator, that is, suppose we have an element of* $(\Pi n : \mathbb{N})\,\mathrm{locator}\,(x_n)$ *. Then we have a locator for* $\lim_{n\to\infty} x_n$*.*

Proof. Let $q < r$ be arbitrary rationals. We need $(q < \lim_{n\to\infty} x_n) + (\lim_{n\to\infty} x_n < r)$. Set $\varepsilon := \frac{r-q}{3}$ so that $q + \varepsilon < r - \varepsilon$. Recall from Remark 4.4.3 that $\left|x_{M(\varepsilon/2)} - \lim_{n\to\infty} x_n\right| < \varepsilon$, that is

$$x_{M(\varepsilon/2)} - \varepsilon < \lim_{n\to\infty} x_n < x_{M(\varepsilon/2)} + \varepsilon.$$

We consider the locator equipped to $x_{M(\varepsilon/2)}$ and do case analysis on $q + \varepsilon <_{x_{M(\varepsilon/2)}} r - \varepsilon$. If we locate $q + \varepsilon < x_{M(\varepsilon/2)}$ then we can locate $q < \lim_{n\to\infty} x_n$. If we locate $x_{M(\varepsilon/2)} < r - \varepsilon$ then we can locate $\lim_{n\to\infty} x_n < r$. □

Remark 6.8.2. We emphasize that Lemma 6.8.1 requires the sequence to be *equipped* with a modulus of Cauchy convergence, whereas existence suffices for the computation of the limit itself, as in Corollary 4.4.7.

Example 6.8.3 (Locators for exponentials)**.** Given a locator for x, we can use Lemma 6.6.1 to obtain a modulus of Cauchy convergence of $\exp(x) = \sum_{k=0}^{\infty} \frac{x^k}{k!}$. Hence $\exp(x)$ has a locator.

Example 6.8.4. Many constants such as π and e have locators, which can be found by examining their construction as limits of sequences.

We can now construct locators for limits of sequences whose elements have locators, and so using Lemma 6.3.1, in particular, limits for sequences of rationals. As we will make precise in Theorem 6.9.7, this covers all the cases.

6.9 Calculating digits

Example 6.9.1. We would like to print digits for numbers equipped with locators, such as π. Such a digit expansion gives rise to rational bounds of the number in question: if a digit expansion of π starts with $3.1\ldots$, then we have the bounds $3.0 < \pi < 3.3$.

We now wish to generate the entire sequence of digits of a real number x equipped with a locator. As in computable analysis and other settings where one works intensionally, with reals given as Cauchy sequences or streams of digits, we wish to extract digit representations from a real equipped with a locator.

In fact, various authors including Brouwer [29] and Turing [94] encountered problems with computing decimal expansions of real numbers in their work. As is common in constructive analysis, we instead consider *signed*-digit representations. Wiedmer [99] shows how to calculate directly on the signed-digit representations in terms of computability theory.

Definition 6.9.2. A signed-digit representation for $x : \mathbb{R}$ is given by $k : \mathbb{Z}$ and a sequence a of signed digits $a_i \in \left\{ \bar{9}, \bar{8}, \ldots, \bar{1}, 0, 1, \ldots, 9 \right\}$, with $\bar{a} :\equiv -a$, such that

$$x = k + \sum_{i=0}^{\infty} a_i \cdot 10^{-i-1}.$$

Example 6.9.3. The number π may be given by a signed-decimal expansion as $3.1415\ldots$, or as $4.\bar{8}\bar{6}15\ldots$, or as $3.2\bar{5}\bar{8}\bar{5}\ldots$.

Lemma 6.9.4. *For any x equipped with a locator, we can find $k : \mathbb{Z}$ such that $x \in (k-1, k+1)$.*

Proof. Use Lemma 6.6.3 with $\varepsilon = 1$ to obtain rationals $u < v$ with $u < x < v$ and $v < 1 + u$. Set $k = \lfloor u \rfloor + 1$. Then:

$$k - 1 = \lfloor u \rfloor \le u < x < v < u + 1 < k + 1. \qquad \square$$

Theorem 6.9.5. *For a real number x, locators and signed-digit representations are interdefinable.*

Proof. If a real number has a signed-digit representation, then it is the limit of a sequence of rational numbers, and so by Lemma 6.8.1 it has a locator.

Conversely, assume a real x has a locator. By Lemma 6.9.4 we get $k : \mathbb{Z}$ with $x \in (k-1, k+1)$. Consider the equidistant subdivision

$$k - 1 < k - \frac{9}{10} < \cdots < k - \frac{1}{10} < k < k + \frac{1}{10} < \cdots < k + 1.$$

By applying the locator several times, we can find a signed digit a_0 such that

$$k + \frac{a_0 - 1}{10} < x < k + \frac{a_0 + 1}{10}.$$

We find subsequent digits in a similar way. □

Note that since $\mathbb{R}$ is Cauchy complete, there is a canonical inclusion $\mathbb{R}_C \to \mathbb{R}$ from the Cauchy reals into $\mathbb{R}$.

Definition 6.9.6. We write isCauchyReal(x) for the claim that a given real $x : \mathbb{R}$ is in the image of the canonical inclusion of the Cauchy reals into $\mathbb{R}$. Equivalently, isCauchyReal(x) holds when there is a rational Cauchy sequence with limit x.

We emphasize that $\|\text{locator}(x)\|$ is *not* equivalent to the locatedness property of Definition 4.6.1.

Theorem 6.9.7. *The following are equivalent for $x : \mathbb{R}$:*

1. $\|\text{locator}(x)\|$, *that is, there exists a locator for x.*

2. *There exists a signed-digit representation of x.*

3. *There exists a Cauchy sequence of rationals that x is the limit of.*

4. isCauchyReal(x).

Proof. Items 1 and 2 are equivalent by Theorem 6.9.5. Item 2 implies item 3 since a signed-digit representation gives rise to a sequence with a modulus of Cauchy convergence. Item 3 implies item 1 because a sequence of rational numbers with modulus of Cauchy convergence has a locator by Lemma 6.8.1. Equivalence of items 3 and 4 is a standard result. □

Remark 6.9.8. The notion of locator can be truncated into a proposition in three ways:

$$\|(\Pi q, r : \mathbb{Q}) q < r \to (q < x) + (x < r)\| \tag{6.1}$$

$$(\Pi q, r : \mathbb{Q})\, \|q < r \to (q < x) + (x < r)\| \tag{6.2}$$

$$(\Pi q, r : \mathbb{Q}) q < r \to \|(q < x) + (x < r)\| \tag{6.3}$$

Now (6.1) is $\|\text{locator}(x)\|$, and (6.3) is the locatedness property of Definition 4.6.1, which holds for all $x : \mathbb{R}$ as mentioned in Section 6.1. In summary, we have

$$(6.1) \implies (6.2) \iff (6.3)$$

where the implications to the right can be shown using the induction rule for propositional truncations, and the implication to the left follows from the fact that $q < r$ is a decidable proposition for $q, r : \mathbb{Q}$.

In other words, we cannot expect to be able to equip every real with a locator, as this would certainly imply that the Cauchy reals and the Dedekind reals coincide, which is not true in general [68].

Corollary 6.9.9. *The following are equivalent:*

1. *For every Dedekind real there exists a signed-digit representation of it.*
2. *The Cauchy reals and the Dedekind reals coincide.*

The types $\mathbb{R}_\text{C}$ and $\mathbb{R}_\text{D}$ do not coincide in general, but they do assuming excluded middle or countable choice. We are not aware of a classical principle that is equivalent with the coincidence of $\mathbb{R}_\text{C}$ and $\mathbb{R}_\text{D}$.

6.10 Dedekind cuts structure

Let $x = (L, U)$ be a pair of predicates on the rationals, i.e. $L, U : \mathcal{P}\mathbb{Q}$. In Definition 4.6.1 we specified the necessary *properties* for x to be a Dedekind cut. More explicitly, we have $\mathrm{isCut} : \mathcal{P}\mathbb{Q} \times \mathcal{P}\mathbb{Q} \to \mathrm{HProp}$ defined by:

$$\begin{aligned}
\mathrm{isCut}(x) &\coloneqq \ \mathrm{boundedLower}(x) \wedge \mathrm{boundedUpper}(x) \\
&\quad \wedge \mathrm{closedLower}(x) \wedge \mathrm{closedUpper}(x) \\
&\quad \wedge \mathrm{openLower}(x) \wedge \mathrm{openUpper}(x) \\
&\quad \wedge \mathrm{transitive}(x) \wedge \mathrm{located}(x), \\
\mathrm{boundedLower}(x) &\coloneqq (\exists q : \mathbb{Q}) q < x, \\
\mathrm{boundedUpper}(x) &\coloneqq (\exists r : \mathbb{Q}) x < r, \\
\mathrm{closedLower}(x) &\coloneqq (\forall q, q' : \mathbb{Q})(q < q') \wedge (q' < x) \Rightarrow q < x, \\
\mathrm{closedUpper}(x) &\coloneqq (\forall r, r' : \mathbb{Q})(r' < r) \wedge (x < r') \Rightarrow x < r, \\
\mathrm{openLower}(x) &\coloneqq (\forall q : \mathbb{Q}) q < x \Rightarrow (\exists q' : \mathbb{Q})(q < q') \wedge (q' < x), \\
\mathrm{openUpper}(x) &\coloneqq (\forall r : \mathbb{Q}) x < r \Rightarrow (\exists r' : \mathbb{Q})(r' < r) \wedge (x < r'), \\
\mathrm{transitive}(x) &\coloneqq (\forall q, r : \mathbb{Q})(q < x) \wedge (x < r) \Rightarrow (q < r), \\
\mathrm{located}(x) &\coloneqq (\forall q, r : \mathbb{Q})(q < r) \Rightarrow (q < x) \vee (x < r).
\end{aligned}$$

We may also consider when x has these data as *structure*, that is, when it is equipped with the structure $\mathrm{isCut}^{\S} : \mathcal{P}\mathbb{Q} \times \mathcal{P}\mathbb{Q} \to \mathcal{U}$ defined by:

$$\begin{aligned}
\mathrm{isCut}^{\S}(x) &\coloneqq \ \mathrm{boundedLower}^{\S}(x) \times \mathrm{boundedUpper}^{\S}(x) \\
&\quad \times \mathrm{closedLower}^{\S}(x) \times \mathrm{closedUpper}^{\S}(x) \\
&\quad \times \mathrm{openLower}^{\S}(x) \times \mathrm{openUpper}^{\S}(x) \\
&\quad \times \mathrm{transitive}^{\S}(x) \times \mathrm{located}^{\S}(x),
\end{aligned}$$

$$\begin{aligned}
\mathrm{boundedLower}^{\S}(x) &\coloneqq (\Sigma q : \mathbb{Q}) q < x, \\
\mathrm{boundedUpper}^{\S}(x) &\coloneqq (\Sigma r : \mathbb{Q}) x < r, \\
\mathrm{closedLower}^{\S}(x) &\coloneqq (\Pi q, q' : \mathbb{Q})(q < q') \times (q' < x) \to q < x, \\
\mathrm{closedUpper}^{\S}(x) &\coloneqq (\Pi r, r' : \mathbb{Q})(r' < r) \times (x < r') \to x < r, \\
\mathrm{openLower}^{\S}(x) &\coloneqq (\Pi q : \mathbb{Q}) q < x \to (\Sigma q' : \mathbb{Q})(q < q') \times (q' < x), \\
\mathrm{openUpper}^{\S}(x) &\coloneqq (\Pi r : \mathbb{Q}) x < r \to (\Sigma r' : \mathbb{Q})(r' < r) \times (x < r'), \\
\mathrm{transitive}^{\S}(x) &\coloneqq (\Pi q, r : \mathbb{Q})(q < x) \times (x < r) \to (q < r), \\
\mathrm{located}^{\S}(x) &\coloneqq (\Pi q, r : \mathbb{Q})(q < r) \to (q < x) + (x < r) = \mathrm{locator}(x).
\end{aligned}$$

In this section we investigate when $x = (L, U)$ has the property $\mathrm{isCut}(x)$, and when it has the data $\mathrm{isCut}^{\S}(x)$. First, note that we cannot expect all Dedekind cuts to come equipped with that data.

Lemma 6.10.1. *Suppose given a choice* $(\Pi x : \mathbb{R})\, \mathrm{locator}(x)$ *of locator for each* $x : \mathbb{R}$*. Then we can define a strongly non-constant function* $f : \mathbb{R} \to \mathbf{2}$ *in the sense that there exist reals* $x, y : \mathbb{R}$ *with* $f(x) \neq f(y)$*.*

Proof. Given a locator for $x : \mathbb{R}$, we can output true or false depending on whether the locator return the left or the right summand for $0 < 1$, as follows.

$$f(x) = \begin{cases} \mathrm{true} & \text{if } \mathrm{locatesRight}(0 <_x 1) \\ \mathrm{false} & \text{if } \mathrm{locatesLeft}(0 <_x 1). \end{cases}$$

The map thus constructed must give a different answer for the real numbers 0 and 1. □

Since any strongly non-constant map from the reals to the Booleans gives rise to a discontinuous map on the reals, we have violated the continuity principle that every map on the reals is continuous. Following Ishihara [55], we can derive WLPO from it.

Lemma 6.10.2. *If there exists a strongly non-constant function* $\mathbb{R} \to \mathbf{2}$*, then* WLPO *holds.*

Proof. Since WLPO is a proposition, we may assume to have $f : \mathbb{R} \to \mathbf{2}$ and $x, y : \mathbb{R}$ with $f(x) \neq f(y)$. Let $P : \mathbb{N} \to \mathrm{DHProp}$ be a decidable predicate.

We start by setting up a decision procedure. We define two sequences $a, b : \mathbb{N} \to \mathbb{R}$ with $f(a_i) = \mathtt{ff}$ and $f(b_i) = \mathtt{tt}$ for each i, and so that a and b converge to the same real l.

Without loss of generality, assume $f(x) = \mathtt{ff}$ and $f(y) = \mathtt{tt}$, and set:

$$a_0 \coloneqq x \qquad\qquad b_0 \coloneqq y$$

$$a_{n+1} \coloneqq \begin{cases} \frac{a_n+b_n}{2} & \text{if } f\left(\frac{a_n+b_n}{2}\right) = \mathtt{ff} \\ a_n & \text{otherwise} \end{cases} \qquad b_{n+1} \coloneqq \begin{cases} \frac{a_n+b_n}{2} & \text{if } f\left(\frac{a_n+b_n}{2}\right) = \mathtt{tt} \\ b_n & \text{otherwise} \end{cases}$$

In words, with a_n and b_n defined, we decide the next point by considering f evaluated at the midpoint $\frac{a_n+b_n}{2}$, and correspondingly updating one of the points. The sequences converge to the same point l. Without loss of generality, we have $f(a_n) = f(l) = \mathtt{ff}$ and $f(b_n) = \mathtt{tt}$ for all $n : \mathbb{N}$.

We may now decide $\neg(\exists n : \mathbb{N})P(n)$. We first define a sequence $c : \mathbb{N} \to \mathbb{R}$ as follows. For a given $n : \mathbb{N}$, we decide if there is any $i < n$ for which $P(i)$ holds, and if so, we set $c_n = b_i$ for the least such i. Otherwise, we set $c_n = l$.

The sequence c converges, giving a limit $m : \mathbb{R}$. Consider $f(m)$.

If $f(m) = \mathtt{ff}$, then $\neg(\exists n : \mathbb{N})P(n)$, since if there did exist n with $P(n)$, then $m = b_i$ for some $i \leq n$, so that $f(m) = f(b_i) = \mathtt{tt}$.

If $f(m) = \mathtt{tt}$, then $\neg\neg(\exists n : \mathbb{N})P(n)$, since if $(\forall n : \mathbb{N})\neg P(n)$ then $m = l$ and so $f(m) = \mathtt{ff}$. □

The following key theorem explains the relationships between being a Dedekind cut, having the Dedekind data $\mathrm{isCut}^{\S}(x)$, and equipping a real with a locator.

Theorem 6.10.3. *For a pair $x = (L, U)$ of predicates on the rationals we have the following:*

1. $\mathrm{isCut}^{\S}(x) \to \mathrm{isCut}(x)$,

2. $\left\|\mathrm{isCut}^{\S}(x)\right\| \Rightarrow \mathrm{isCut}(x)$,

3. $\mathrm{isCut}(x) \times \mathrm{locator}(x) \to \mathrm{isCut}^{\S}(x)$,

4. $\mathrm{isCut}(x) \times \|\mathrm{locator}(x)\| \Rightarrow \mathrm{isCauchyReal}(x)$, *and*

5. $\left\|\mathrm{isCut}^{\S}(x)\right\| \Rightarrow \mathrm{isCauchyReal}(x)$.

The third item tells us that for a given Dedekind real x, in order to obtain the structures that make up $\mathrm{isCut}^{\S}(x)$, we only require $\mathrm{locator}(x)$.

Proof. We show the first item by considering all property/structure-pairs above.

$\mathrm{boundedLower}^{\S}(x) \to \mathrm{boundedLower}(x)$ follows by applying the truncation map $|\cdot|$ of Definition 2.4.3, and similarly for boundedUpper.

$\mathrm{closedLower}^{\S}(x) \to \mathrm{closedLower}(x)$ is trivial since, following Definition 2.4.5, their definitions work out to the same thing: we do not need to make any changes to make $\mathrm{closedLower}^{\S}$ structural.

$\mathrm{openLower}^{\S}(x) \to \mathrm{openLower}(x)$ by a pointwise truncation: let $q : \mathbb{Q}$ be arbitrary and assume $q < x$, then we get $(\Sigma q' : \mathbb{Q})(q < q') \times (q' < x)$, and hence $(\exists q' : \mathbb{Q})(q < q') \wedge (q' < x)$.

Again following Definition 2.4.5, $\mathrm{transitive}(x)$ and $\mathrm{transitive}^{\S}(x)$ are defined equally.

$\mathrm{locator}(x) \to \mathrm{located}(x)$ again by a pointwise truncation.

The second item follows using the elimination rule for propositional truncations since $\mathrm{isCut}(x)$ is a proposition.

For the third item, it remains to construct bounds, and to construct $\mathrm{openLower}^{\S}(x)$ and $\mathrm{openUpper}^{\S}(x)$. The former is Lemma 6.6.1. The latter follows from the Archimedean structure of Lemma 6.7.3 and the fact that we have locators for rationals, as in Lemma 6.3.1.

The fourth item follows from Theorem 6.9.7.

The fifth item follows by combining the second and the fourth. □

Remark 6.10.4. Indeed, as already touched upon in Remark 6.6.4, the above means that, rather than only equipping *locators* to our a real number x, we may as well equip the full structure $\mathrm{isCut}^{\S}(x)$, and our work with locators would still go through. This would give the possibility to choose more efficient algorithms to compute with. The point of the above theorem is that just having a locator already suffices.

Theorem 6.10.5. *For an arbitrary pair $x = (L, U)$ of predicates on the rationals it is not provable that* $\mathrm{isCut}(x)$ *implies* $\left\|\mathrm{isCut}^{\S}(x)\right\|$.

Proof. By Theorem 6.10.3, $\left\|\mathrm{isCut}^{\S}(x)\right\|$ implies that x is a Cauchy real. However, in general the Cauchy reals and the Dedekind reals do not coincide [68]. □

6.11 Notes

We have introduced the term *locator* to mean the structure that is the focus of this chapter, and have introduced a basic theory of locators. The fact that the results about locators have equivalents in terms of intensional representations of reals suggests that we are not doing anything new. This is desirable: we merely introduced a particular representation that seems suitable for computation.

Whereas inequalities such as $q < x$ are undecidable, we *can* decide what the output of a locator is: we simply evaluate a locator on rationals $q < r$ and observe what output value we get in the sense of Lemma 6.4.6, and this decidability is exploited in our development. Sections 6.4 and 6.5 give the central techniques.

In order to construct locators for algebraic operations, we first characterized those operations in terms of $<$ in Lemma 6.7.1. This lemma shows in detail a claim left as an exercise in The Univalent Foundations Program [91, Theorem 11.2.14]. We are also led to consider certain strengthenings of the field structure, such as the Archimedean structure in Lemma 6.7.3, strong cotransitivity in Lemma 6.7.4 and Remark 6.7.5, and a structural characterization of algebraic operations in terms of $<$ in Remark 6.7.5. These structural characterizations of field structure are new to our knowledge. They allow us to upgrade mathematical arguments to computational techniques.

Sections 6.8 and 6.9 then establish that locators are simply an alternative representation of rational Cauchy sequences—again desirable as it shows that we are not doing anything fundamentally new. The results can be interpreted as explaining what the difference between the Cauchy and Dedekind reals is: namely, Cauchy reals are those that arise during computation.

Finally, in Section 6.10 we explain why the particular property of a Dedekind cut being located was strengthened into structure: the other properties are automatically also strengthened into structure. This again shows that, in principle, all we need to compute with real numbers is a locator.

Chapter 7

SOME CONSTRUCTIVE ANALYSIS WITH LOCATORS

We show some ways of using locators in an existing theory of constructive analysis on a pre-defined set of reals. This means that our work does not depend on other results being developed in terms of some chosen representation of real numbers: locators can be added after the fact.

The central notion is that of functions on the reals that *lift to locators*, discussed in Section 7.1, which is neither weaker nor stronger than continuity. We compute locators for integrals in Section 7.2. We discuss how locators can help computing roots of functions in Section 7.3.

As in Chapter 6, we work with an arbitrary but fixed type of reals $\mathbb{R}$, namely any Cauchy complete Archimedean field in the sense of Chapter 4.

7.1 Preliminaries

What are the functions on the reals that allow us to compute? When such a function $f : \mathbb{R} \to \mathbb{R}$ is applied to an input real number $x : \mathbb{R}$ that we can compute with, then we should be able to compute with the output $f(x)$. This can be formalized in terms of locators in the following straightforward way, which we use as an abstract notion of computation.

Definition 7.1.1. A function $f : \mathbb{R} \to \mathbb{R}$ *lifts to locators* if it comes equipped with a method for constructing a locator for $f(x)$ from a locator for x. This means that f lifts to locators if it is equipped with an element of the type

$$(\Pi x : \mathbb{R})\, \mathrm{locator}(x) \to \mathrm{locator}(f(x)).$$

Another way to say this is that f lifts to locators iff we can find the top edge in the diagram

$$\begin{tikzcd}
\mathbb{R}^{\mathfrak{L}} \arrow[r] \arrow[d, "\mathrm{pr}_1"] \arrow[dr, phantom, "\circ"] & \mathbb{R}^{\mathfrak{L}} \arrow[d, "\mathrm{pr}_1"] \\
\mathbb{R} \arrow[r, "f"] & \mathbb{R}
\end{tikzcd}$$

where $\mathbb{R}^{\mathfrak{L}} :\equiv (\Sigma x : \mathbb{R})\, \mathrm{locator}(x)$ is the type of real numbers equipped with locators.

“Lifting to locators” itself is structure.

Remark 7.1.2. If the reals are defined intensionally, for example as the collection of all Cauchy sequences without quotienting, then every function on them is defined completely by its behavior on those intensional reals. However, in our case, given only the lifting structure $\mathbb{R}^{\mathfrak{L}} \to \mathbb{R}^{\mathfrak{L}}$, we cannot recover the function $f : \mathbb{R} \to \mathbb{R}$, because we do not have a locator for every $x : \mathbb{R}$.

In other words, well-behaved maps are specified by two pieces of data, namely a function $f : \mathbb{R} \to \mathbb{R}$ representing the extensional value of the function, and a map $\mathbb{R}^{\mathfrak{L}} \to \mathbb{R}^{\mathfrak{L}}$ that tells us how to compute.

Example 7.1.3. The exponential function $\exp(x) = \sum_{k=0}^{\infty} \frac{x^k}{k!}$ of Examples 4.4.8 and 6.8.3 lifts to locators, for example using our construction of locators for limits as in Lemma 6.8.1.

In order to start developing analysis, we define some notions of continuity.

Definition 7.1.4. A function $f : \mathbb{R} \to \mathbb{R}$ is *continuous at* $x : \mathbb{R}$ if

$$(\forall \varepsilon : \mathbb{Q}_+)(\exists \delta : \mathbb{Q}_+)(\forall y : \mathbb{R})\, |x - y| < \delta \Rightarrow |f(x) - f(y)| < \varepsilon.$$

f is *pointwise continuous* if it is continuous at all $x : \mathbb{R}$.

Definition 7.1.5. A *modulus of uniform continuity for* f *on* $[a, b]$, with $a, b : \mathbb{R}$, is a map $\omega : \mathbb{Q}_+ \to \mathbb{Q}_+$ with:

$$(\forall x, y \in [a, b])\, |x - y| < \omega(\varepsilon) \Rightarrow |f(x) - f(y)| < \varepsilon.$$

Example 7.1.6 (Continuity of exp)**.** For any a, b, there *exists* a modulus of uniform continuity for exp on $[a, b]$: after all, if $|x - y| < \frac{\varepsilon}{\exp(1+\max(0,a,b)))}$ then we have $|\exp(x) - \exp(y)| < \varepsilon$. The existence of a modulus of uniform continuity then follows since we know that there exists a rational $E : \mathbb{Q}_+$ such that $E < \frac{1}{\exp(1+\max(0,a,b))}$. If a and b have locators, for instance when they are rationals, then we can *find* a modulus of uniform continuity for exp on that interval by *computing* such a positive lower bound E. This reasoning applies to various functions for which real error bounds can be expressed. This notion of continuity on a subset is made precise later by Definition 8.1.7.

From a constructive viewpoint in which computation and continuity align, it would be desirable if some form of continuity of $f : \mathbb{R} \to \mathbb{R}$ would imply that it lifts to locators. Alas, this is not the case, not even for constant functions.

Lemma 7.1.7. *If it holds that all constant functions lift to locators, then every $x : \mathbb{R}$ comes equipped with a locator.*

Using Lemmas 6.10.1 and 6.10.2, this then yields the constructive taboo WLPO.

Proof. For $x : \mathbb{R}$, let $f : \mathbb{R} \to \mathbb{R}$ be the constant map at x, and note that f is continuous, so that by assumption it lifts to locators. Since the rational number 0 has a locator, $f(0) = x$ has a locator. □

The converse direction, that lifting to locators would imply continuity, also fails dramatically.

Lemma 7.1.8. *Assuming* PEM, *we can define a discontinuous map $f : \mathbb{R} \to \mathbb{R}$ that lifts to locators.*

Proof. We can use PEM to define a discontinuous function, which automatically lifts to locators by applying Lemma 6.1.3. □

It may be the case that the structure of lifting to locators can be used to strengthen certain *properties* of continuity into *structures*. For example, does every function that lifts to locators and is pointwise continuous come equipped with the structure

$$(\Pi x : \mathbb{R})(\Pi \varepsilon : \mathbb{Q}_+)(\Sigma \delta : \mathbb{Q}_+)(\forall y : \mathbb{R})\, |x - y| < \delta \Rightarrow |f(x) - f(y)| < \varepsilon$$

of structural pointwise continuity at every $x : \mathbb{R}$? We leave this as an open question.

For the above reasons, the theorems in this chapter and the next assume continuity *and* a structure of lifting to locators: the former to make the constructive analysis work, and the latter to compute.

7.2 Integrals

We can compute definite integrals of uniformly continuous functions in the following way.

Theorem 7.2.1. *Suppose $f : \mathbb{R} \to \mathbb{R}$ has a modulus of uniform continuity on $[a, b]$, and a and b are real numbers with locators. Suppose that f lifts to locators. Then $\int_a^b f(x)\,\mathrm{d}x$ has a locator.*

Proof. For uniformly continuous functions, the integral $\int_a^b f(x)\,\mathrm{d}x$ can be computed as the limit

$$\lim_{n\to\infty} \frac{b-a}{n} \sum_{k=0}^{n-1} f\left(a + k \cdot \frac{b-a}{n}\right).$$

Now every value

$$\frac{b-a}{n} \sum_{k=0}^{n-1} f\left(a + k \cdot \frac{b-a}{n}\right).$$

in the sequence comes equipped with a locator using Lemmas 6.3.1 and 6.7.6, and using the fact that a and b have locators and f lifts to locators. From the modulus of uniform continuity of f, and the computation of a rational B with $b - a \leq B$ using Lemmas 6.7.6 and 6.6.1 we can compute a modulus of Cauchy convergence of the sequence. Hence the limit has a locator using Lemma 6.8.1. □

Combining this with the calculation of signed-digit representations of reals with locators in Theorem 6.9.5, the above means we can generate the digit sequence of certain integrals. Through the construction of close bounds in Lemma 6.6.3, we can in principle verify the value of integrals up to arbitrary precision.

> *Remark* 7.2.2. Integrals, as elements of $\mathbb{R}$, can be defined given only the *existence* of a modulus of uniform continuity. To get a locator, we use the modulus of uniform continuity to find a modulus of Cauchy convergence.

Example 7.2.3. The integral $\int_0^8 \sin(x + \exp(x))\,\mathrm{d}x$ has a locator (where sin is defined, and shown to lift to locators, in a way similar to exp). This integral is often incorrectly approximated by computer algebra systems. Mahboubi, Melquiond, and Sibut-Pinote [73, Section 6.1] have formally verified approximations of this integral, and in principle our work gives an alternative method to do so. However, our constructions are not efficient enough to do so in practice, and we give some possible remedies in the conclusions in Chapter 10.

7.3 Intermediate value theorems

We may often compute locators of real numbers simply by analysing the proof of existing theorems in constructive analysis. The following construction of the root of a function is an example of us being able to construct locators simply by following the proof in the literature.

Theorem 7.3.1. *Suppose f is pointwise continuous on the interval $[a, b]$ and $f(a) < 0 < f(b)$ with $a, b : \mathbb{R}$. Then for every $\varepsilon : \mathbb{Q}_+$ we can find $x : \mathbb{R}$ with $|f(x)| < \varepsilon$. If f lifts to locators, and a and b are equipped with locators, then x is equipped with a locator.*

Proof. We define sequences $c, d, z, w : \mathbb{N} \to \mathbb{R}$ as in Frank [45]:

$$\begin{aligned} z_0 &= a & c_n &= (z_n + w_n)/2 & z_{n+1} &= c_n - d_n(b-a)/2^{n+1} \\ w_0 &= b & d_n &= \max\left(0, \min\left(\frac{1}{2} + \frac{f(c_n)}{\varepsilon}, 1\right)\right) & w_{n+1} &= w_n - d_n(b-a)/2^{n+1} \end{aligned}$$

with x defined as the limit of $c : \mathbb{N} \to \mathbb{R}$, which converges since $z, w : \mathbb{N} \to \mathbb{R}$ are monotone sequences with $z_n \le c_n \le w_n$ and $z_n - w_n = (b-a)/2^n$. The fact that $|f(x)| < \varepsilon$ can be shown as in Frank [45]. Because f lifts to locators, and a and b have a locator, all c_n have locators. For a modulus of Cauchy convergence, Lemma 6.7.6 gives a locator for $b - a$ so that we can use Lemma 6.6.1 to compute a rational B with $|z_n - w_n| \le B/2^n$. So by Lemma 6.8.1, x has a locator. □

We will now work towards an intermediate value theorem in which the locators help us with the computation of the root itself, avoiding any choice principles. We stated this intermediate value theorem and its proof informally in the introduction to Chapter 6.

Definition 7.3.2. A function $f : \mathbb{R} \to \mathbb{R}$ is *locally nonconstant* if for all $x < y$ and $t : \mathbb{R}$, there exists $z : \mathbb{R}$ with $x < z < y$ and $f(z) \# t$, recalling that $(f(z) \# t) = (f(z) > t) \vee (f(z) < t)$.

Example 7.3.3. Every strictly monotone function is locally nonconstant, but not every locally nonconstant function is strictly monotone.

Lemma 7.3.4. *Suppose f is a pointwise continuous function, and x, y and t are real numbers with locators with $x < y$. Further suppose that f is locally nonconstant, and lifts to locators. Then we can find $r : \mathbb{Q}$ with $x < r < y$ and $f(r) \# t$.*

Proof. Since f is locally nonconstant, there *exist* $z : \mathbb{R}$ and $\varepsilon : \mathbb{Q}_+$ with $|f(z) - t| > \varepsilon$. Since f is continuous at z, there exists $q : \mathbb{Q}$ with $|f(q) - t| > \varepsilon/2$. Since $\mathbb{Q}_+$ and $\mathbb{Q}$ are denumerable, we can find $r : \mathbb{Q}$ such that there exists $\eta : \mathbb{Q}_+$ with $|f(r) - t| > \eta$. In particular r satisfies $|f(r) - t| > 0$, that is, $f(r) \# t$. □

The above result can be thought of as saying that if f is a pointwise continuous function that lifts to locators, then the *property* of local nonconstancy implies a certain *structure* of local nonconstancy: for given reals with locators $x < y$ and t, we do not just get the existence of a real z, but we can explicitly choose a point z where f is apart from t.

Exact intermediate value theorems based on local nonconstancy usually assume dependent choice, see e.g. Bridges and Richman [28, Chapter 3, Theorem 2.5] or Troelstra and van Dalen [92, Chapter 6, Theorem 1.5]. The following result holds in the absence of such choice principles. It can perhaps be compared to developments in which the real numbers are represented directly as Cauchy sequences [86, 87, 51] or with Taylor [90]. Note, however, that

1. we assume local nonconstancy rather than monotonicity, and that

2. we use the *property* of local nonconstancy to compute roots, rather than assuming this as structure.

Theorem 7.3.5. *Suppose f is a pointwise continuous function, and $a < b$ are real numbers with locators. Further suppose that f is locally nonconstant, and lifts to locators, with $f(a) \leq 0 \leq f(b)$. Then we can find a root of f, which comes equipped with a locator.*

Proof. We define sequences $a, b : \mathbb{N} \to \mathbb{R}$ with $a_n < a_{n+1} < b_{n+1} < b_n$, with $f(a_n) \leq 0 \leq f(b_n)$, with $b_n - a_n \leq (b-a)\left(\frac{2}{3}\right)^n$, and such that all a_n and b_n have locators. Set $a_0 = a$, $b_0 = b$. Suppose a_n and b_n are defined, and use Lemma 7.3.4 to find q_n with $\frac{2a_n+b_n}{3} < q_n < \frac{a_n+2b_n}{3}$ and $f(q_n) \# 0$.

- If $f(q_n) > 0$, then set $a_{n+1} :\equiv a_n$ and $b_{n+1} :\equiv q_n$.
- If $f(q_n) < 0$, then set $a_{n+1} :\equiv q_n$ and $b_{n+1} :\equiv b_n$.

For a modulus of Cauchy convergence, we can compute a locator for $b - a$ and from this we can compute a rational B with $|b_n - a_n| \leq B\left(\frac{2}{3}\right)^n$. The sequences converge to a number x. For any ε, we have $|f(x)| \leq \varepsilon$, hence $f(x) = 0$. □

> *Remark* 7.3.6. Since we only appealed to Lemma 7.3.4 with $t = 0$, that is, since we were only interested in points where f is apart from 0, Theorem 7.3.5 may be strengthened by only requiring that f is locally nonzero.

Example 7.3.7. The function exp is strictly increasing, and hence locally nonconstant. So if $y > 0$ has a locator, then $\exp(x) = y$ has a solution x with a locator.

7.4 Notes

Many proofs in this chapter have intentionally been written as similar as possible to originals from constructive and traditional mathematics. The presence of the locators is not to make the constructive analysis work; rather, it is to make the computation work. In this sense, we have made the computation work without a conceptual burden of intensional representations.

The new notion of *lifting to locators* grew out of a naive desire to have locators for the output of a function whenever we have a locator for the input. We have left the following open question: given that $f : \mathbb{R} \to \mathbb{R}$ lifts to locators, do we obtain a certain *structure* of continuity from a *property* of continuity?

We have not spent much time finding an alternative notion of "functions that compute" with a closer relationship to continuity, and this could be the topic of further research. Such a notion could perhaps allow for more satisfying formulations of the theorems in Sections 7.2 and 7.3.

Theorem 7.3.5 is an improvement on existing exact intermediate value theorems [86, 90] since it assumes the *property* of local nonconstancy to compute roots.

Chapter 8

METRIC SPACES

Uniqueness of limits, a basic result of metric spaces, crucially makes use of the property that $\rho(x, y) = 0 \Rightarrow x = y$. When the points of the space are equipped with further non-unique data, such as when real numbers are equipped with locators, this property is no longer satisfied, so that we obtain a *pseudometric* space, and some results such as uniqueness of limits have to be rephrased to conclude $\rho(x, y) = 0$ instead of $x = y$. Spaces in which zero distance does not imply equality have also been considered in Bourbaki [27, Chapter IX].

Remark 8.0.1. *Pseudo*metric spaces are distinct from the more general *pre*metric spaces from Section 4.5.

We study both pseudometric and metric spaces, and will work towards solutions of differential equations in Section 8.4, with the reals equipped with locators, and the functions involved lifting them, so that we can compute a solution of differential equations that gives us discrete data.

We start by defining (pseudo)metric spaces and giving their basic theory, including convergence and uniform continuity, in Section 8.1. Then we consider Lipschitz continuous endomaps on a (pseudo)metric space, and the construction of their fixpoints, in Section 8.3. We construct a particular endomap in Section 8.4 that computes Picard iterations for a differential equation, on a pseudometric space of uniformly continuous maps $[a, b] \to [c, d]$ that lift to locators. In this way, we compute solutions for differential equations as limits of certain sequences. Because this solution lifts to locators, we can compute the solutions of such differential equations and produce, for instance, signed-digit representations.

The majority of this chapter does not contain substantially new proofs. Instead, the majority of the proofs have been developed as they have been in existing literature. This may mislead some readers to conclude that nothing new has been added. However, the statements of the theorems have been changed. For instance, the solution of differential equations in Theorem 8.4.7 is a function that *lifts to locators* in an appropriate sense, while the proof is essentially a traditional Picard iteration. In this way, the traditional proof techniques can be used to give discrete data out of solutions to differential equations, such as digit approximations at certain inputs.

We emphasize, as we did in Chapters 6 and 7, that the purpose of the work is to show *how* to extract discrete information from real analysis in UTT, as opposed to arguing which system is most suited to such computation. There are three main differences with traditional mathematics:

1. the use of type theory rather than set theory,
2. the heavy use of moduli of uniform continuity for reasons of constructivity, and
3. the use of locators on top of a pre-supplied development with real numbers rather than computing solely with reals by representations.

Throughout this chapter, we fix a set of reals $\mathbb{R}$, namely any Cauchy complete Archimedean field in the sense of Chapter 4. We have the set $\mathbb{R}_{\geq 0}$ of nonnegative reals, $\mathbb{R}_{+}$ is the set of positive reals, and we identify their elements with their underlying element of $\mathbb{R}$.

8.1 Basic definitions in pseudometric spaces

The basic definitions of metric spaces are inspired by Bishop and Bridges [20].

Definition 8.1.1. For a set X, a *metric* is a map $\rho : X \times X \to \mathbb{R}_{\geq 0}$ satisfying, for $x, y, z : X$:

1. $\rho(x, x) = 0$, and
2. $\rho(x, y) = \rho(y, x)$, and

3. $\rho(x,z) \leq \rho(x,y) + \rho(y,z)$, and

4. $\rho(x,y) = 0 \Rightarrow x = y$.

A *pseudometric* is a map $\rho : X \times X \longrightarrow \mathbb{R}_{\geq 0}$ that only satisfies conditions 1–3.

> *Remark* 8.1.2. The first condition is equivalent to $x = y \Rightarrow \rho(x,y) = 0$.

The real numbers are an important example of a metric space.

Lemma 8.1.3. *The Euclidean distance $\rho(x,y) = |x - y|$ is a metric on $\mathbb{R}$.*

Example 8.1.4. Of particular interest is the collection of elements of uniformly continuous functions $[a,b] \longrightarrow [c,d]$ equipped with the structure of lifting to locators. We discuss this in more detail in Section 8.2.

Whenever we have a (pseudo)metric on a set X, we canonically set the premetric $\sim$ to be, for $x, y : X$ and $\varepsilon : \mathbb{Q}_+$:

$$x \sim_\varepsilon y :\equiv \rho(x,y) < \varepsilon.$$

The notions of *Cauchy approximation*, convergence to a *limit*, and *Cauchy completeness* are thus inherited from the premetric as in Section 4.5. In Section 8.4, we compute solutions of differential equations as limits of Cauchy sequences.

Lemma 8.1.5. *In a pseudometric space, if a sequence x has points x_∞ and y_∞ as limits, then $\rho(x_\infty, y_\infty) = 0$. Hence, in metric spaces, limits are unique.*

Proof. First, because in pseudometric spaces the premetric is transitive and symmetric in the expected way, we have $(\forall \varepsilon : \mathbb{Q}_+) x_\infty \sim_\varepsilon y_\infty$. Hence we have $\rho(x_\infty, y_\infty) \leq 0$, and so in metric spaces we get $x_\infty = y_\infty$. □

The lack of uniqueness of limits in pseudometric spaces means that, from existence of limits, we do not obtain a map that computes limits. So we require Cauchy complete pseudometric spaces to have such a map directly, and we do not have an equivalent of Corollary 4.4.6 for pseudometric spaces.

Additionally, theorems that compute certain limits in pseudometric spaces have a *structure* as their conclusion, rather than a *proposition.* This means that, generally speaking, such theorems need stronger assumptions than their equivalents for metric spaces.

Definition 8.1.6. A pseudometric space (X, ρ) is *Cauchy complete* when it has a lim function which, given a sequence with Cauchy modulus, computes a limit of it. In other words, (X, ρ) is Cauchy complete when from a sequence equipped with a modulus we can obtain a limit.

In our constructive setting, there is no known general notion of continuity of maps between metric spaces that makes $1/x$ continuous on the reals apart from 0 and is closed under composition [85]. We avoid this issue by working with uniformly continuous functions on finite intervals of real numbers, where uniform continuity is defined in terms of the premetric. We define this simply by generalizing Definition 7.1.5 to pseudometric spaces.

Definition 8.1.7. For premetric spaces $(X, \sim)$ and $(Y, \sim)$, a *modulus of uniform continuity* of a map $f : X \to Y$ is an assignment $\omega : \mathbb{Q}_+ \to \mathbb{Q}_+$ such that

$$(\forall \varepsilon : \mathbb{Q}_+)(\forall x, y : X) x \sim_{\omega(\varepsilon)} y \Rightarrow f(x) \sim_\varepsilon f(y).$$

For pseudometric spaces (X, ρ) and (Y, σ), this condition states

$$(\forall \varepsilon : \mathbb{Q}_+)(\forall x, y : X) \rho(x, y) < \omega(\varepsilon) \Rightarrow \sigma(f(x), f(y)) < \varepsilon.$$

And in particular for the Euclidean distance of Lemma 8.1.3 this coincides with Definition 7.1.5.

As usual, the identity map has a modulus of uniform continuity (namely $\omega(\varepsilon) = \varepsilon$), and we can compose uniformly continuous maps in the sense that if $f : X \to Y$ has modulus ω and $g : Y \to Z$ has ψ, then $g \circ f$ has modulus $\omega \circ \psi$. Note we use the reverse composition for the modulus.

Uniformly continuous functions preserve Cauchy sequences in the following sense.

Lemma 8.1.8. *Let (X, ρ) and (Y, σ) be pseudometric spaces. If $f : X \to Y$ has modulus of uniform continuity ω, and a sequence x in X is Cauchy with modulus M, then $\lambda n.f(x_n)$ is Cauchy with modulus $M \circ \omega$.*

Proof. Let $\varepsilon : \mathbb{Q}_+$ be arbitrary, and let $m, n \geq M(\omega(\varepsilon))$. Because M is a modulus of Cauchy convergence for x, we have $\rho(x_m, x_n) < \omega(\varepsilon)$ and hence by the uniform continuity of f, we have $\sigma(f(x_m), f(x_n)) < \varepsilon$. □

Definition 8.1.9. A map $f : X \to Y$ between pseudometric spaces (X, ρ) and (Y, σ) *preserves limits* if whenever x_∞ is a limit of $x : \mathbb{N} \to X$, we also have that $f(x_\infty)$ is a limit of $\lambda n.f(x_n)$. Notice that this limit is not required to be the limit given by the limit operator obtained from Cauchy completeness.

Lemma 8.1.10. *If $f : X \to Y$ between two pseudometric spaces (X, ρ) and (Y, σ) has modulus of uniform continuity ω then it preserves limits.*

Proof. Let x_∞ be a limit of the sequence x. We show that $f(x_\infty)$ is a limit of $\lambda n.f(x_n)$. Let $\varepsilon > 0$, then we need to find $N : \mathbb{N}$ such that whenever $n \geq N$ we have $\sigma(f(x_n), f(x_\infty)) < \varepsilon$.

From the fact that x_∞ is a limit of x, there exists $N : \mathbb{N}$ such that whenever $n \geq N$, we have $\rho(x_n, x_\infty) < \omega(\varepsilon)$, and so by uniform continuity of f we also have $\sigma(f(x_n), f(x_\infty)) < \varepsilon$. □

We will use the above theory of (pseudo)metric spaces to compute solutions of differential equations.

8.2 Function spaces

In this section we develop the pseudometric space in which we compute the solution of a given differential equation. Loosely speaking, this is a space of functions from a closed interval to a closed interval, equipped with modulus of uniform continuity, and with the structure of lifting to locators.

First, by reading a closed interval $[a, b]$ as the Σ-type of elements $x : \mathbb{R}$ satisfying $a \leq x \leq b$, we can make sense of maps $[a, b] \to [c, d]$. Then the metric on closed intervals $[a, b]$ is inherited from $\mathbb{R}$ as $\rho((x, \mu), (y, \nu)) :\equiv |x - y|$, and by identifying elements of $[a, b]$ with their underlying element of $\mathbb{R}$ we can simply write $|x - y|$ for the distance between two elements $x, y : [a, b]$.

Definition 8.2.1. For elements $a < b$ and $c < d$ in $\mathbb{R}$, we define the type $U[a,b][c,d]$ of maps $[a,b] \to [c,d]$ equipped with moduli of uniform convergence as

$$\begin{aligned}&(\Sigma f : [a,b] \to [c,d])\\ &\quad(\Sigma\omega : \mathbb{Q}_+ \to \mathbb{Q}_+)(\forall\varepsilon : \mathbb{Q}_+)(\forall x, y : [a,b])\, |x - y| < \omega(\varepsilon) \Rightarrow |f(x) - f(y)| < \varepsilon.\end{aligned}$$

We identify elements of $U[a,b][c,d]$ with their underlying map $[a,b] \to [c,d]$.

Lemma 8.2.2. *For reals $a < b$, we can calculate the supremum of a function $f : [a,b] \to \mathbb{R}$ with modulus of uniform continuity ω as the limit of the sequence*

$$x := \lambda n.\max\left(f\left(a\right), f\left(a + \frac{b-a}{n+1}\right), \ldots, f\left(b\right)\right)$$

which has modulus of Cauchy convergence

$$M(\varepsilon) := \frac{b-a}{\omega(\varepsilon)}.$$

Proof. The function M is a modulus of the sequence because the maxima are taken over values that do not deviate too much. More precisely, if $m, n \geq M(\varepsilon)$, and without loss of generality $m \leq n$, then for every i between 0 and m there exists j between 0 and n such that $\left|i\frac{b-a}{m+1} - j\frac{b-a}{n+1}\right| < \omega(\varepsilon)$ and hence $\left|f\left(a + i\frac{b-a}{m+1}\right) - f\left(a + j\frac{b-a}{n+1}\right)\right| < \varepsilon$, so that $|x_m - x_n| < \varepsilon$.

We can show that the limit l of the sequence x is an upper bound of f by first showing that l is an upper bound of the sequence itself. Then the fact that it is a limit of the sequence can be used to show that it is a minimal upper bound. □

Lemma 8.2.3. *For elements $a < b$ and $c < d$ in $\mathbb{R}$, the type $U[a,b][c,d]$ is a pseudometric space, with the distance $\rho(f,g)$ defined as the supremum of the pointwise Euclidean distance $|f(x) - g(x)|$, computed as in Lemma 8.2.2.*

Proof. Conditions 1 and 2 of Definition 8.1.1 are shown algebraically.

For the third condition: if x is the point in the interval $[a,b]$ for which $|f(x) - h(x)|$ is maximized, then we have $|f(x) - h(x)| \leq |f(x) - g(x)| + |g(x) - h(x)|$. But by definition we have $|f(x) - g(x)| \leq \rho(f,g)$ and $|g(x) - h(x)| \leq \rho(g,h)$ so that indeed $\rho(f,h) \leq \rho(f,g) + \rho(g,h)$. □

What does it mean for a function on a closed interval to lift to locators? First, since we identify elements of closed intervals with their underlying element of $\mathbb{R}$, a locator for $x : [a, b]$ also gives, for every pair of ordered rationals $q < r$ in $\mathbb{Q}$, one of $q < x$ or $x < r$. In particular, those rationals do *not* need to somehow be contained in the interval $[a, b]$.

Definition 8.2.4. With $a, b, c, d : \mathbb{R}$ and $a < b$, $c < d$, a function $f : [a, b] \to [c, d]$ *lifts to locators* if it comes equipped with a method for constructing a locator for $f(x)$ from a locator for x. This means that f lifts to locators if it is equipped with an element of the type

$$(\Pi x : [a, b])\,\mathrm{locator}(x) \to \mathrm{locator}(f(x)).$$

Another way to say this is that f lifts to locators iff we can find the top edge in the diagram

$$\begin{tikzcd}
{[a,b]^{\mathfrak{L}}} \arrow[r] \arrow[d, "\mathrm{pr}_1"] \arrow[dr, phantom, "\circ"] & {[c,d]^{\mathfrak{L}}} \arrow[d, "\mathrm{pr}_1"] \\
{[a,b]} \arrow[r, "f"] & {[c,d]}
\end{tikzcd}$$

where $[a, b]^{\mathfrak{L}} :\equiv (\Sigma x : [a, b])\,\mathrm{locator}(x)$ is the type of real numbers in the interval $[a, b]$ equipped with locators.

Remark 8.2.5. As was the case for Definition 7.1.1, the notion of lifting to locators is structure rather than property, and we cannot recover the function f from a map $[a, b]^{\mathfrak{L}} \to [c, d]^{\mathfrak{L}}$. The reason for this is that this only gives the values of f for those reals that can be equipped with a locator, but as we saw in Theorem 6.10.3, these are only Cauchy reals.

Lemma 8.2.6. *For any pseudometric space (X, ρ) and any map $\pi : Y \to X$, the map $\sigma(y, z) :\equiv \rho(\pi(y), \pi(z))$ is a pseudometric on Y.*

Definition 8.2.7. For elements $a < b$ and $c < d$ in $\mathbb{R}$, we define the type $UL[a, b][c, d]$ of maps $[a, b] \to [c, d]$ equipped with moduli of uniform convergence, and additionally equipped with the structure of lifting to locators. This may be formalized as the type

$$\begin{aligned}
&(\Sigma f : [a, b] \to [c, d]) \\
&\quad ((\Sigma \omega : \mathbb{Q}_+ \to \mathbb{Q}_+)(\forall \varepsilon : \mathbb{Q}_+)(\forall x, y : [a, b])\, |x - y| < \omega(\varepsilon) \Rightarrow |f(x) - f(y)| < \varepsilon) \\
&\times ((\Pi x : [a, b])\,\mathrm{locator}(x) \to \mathrm{locator}(f(x))).
\end{aligned}$$

Put differently, the type $UL[a,b][c,d]$ consists of triples (f,ω,L) consisting of a function $f : [a,b] \to [c,d]$ with modulus of uniform continuity ω, and the structure L of lifting to locators.

We identify elements of $UL[a,b][c,d]$ with their underlying map $[a,b] \to [c,d]$.

Corollary 8.2.8. *The type $UL[a,b][c,d]$ of maps $[a,b] \to [c,d]$ with modulus of uniform continuity, and the structure of lifting to locators, is a pseudometric space, with distance computed as on $U[a,b][c,d]$.*

We have the pseudometric spaces $U[a,b][c,d]$ and $UL[a,b][c,d]$. While these are not *metric* spaces, they are *almost* metric spaces in the sense that if two functions $f,g : [a,b] \to [c,d]$ have zero distance in these spaces, then those *functions*, as opposed to elements of the pseudometric space, are identical. The original elements of $U[a,b][c,d]$, and similarly $UL[a,b][c,d]$, may be different because they come equipped with different moduli of uniform continuity.

Lemma 8.2.9. *Let (f,ω,L) and (g,ξ,M) be elements of $UL[a,b][c,d]$, and suppose they have distance 0, i.e. are pseudo-equal. Then $f = g$.*

Proof. If f and g have distance 0, then the supremum of their pointwise distances is 0, and hence $|f(x) - g(x)| = 0$ at every point x. This makes f and g pointwise identical, and so by function extensionality identical as functions. □

In summary, we have the following types:

1. the type $[a,b] \to [c,d]$ of arbitrary functions on reals whose inputs and outputs are bounded, on which we do not have any (pseudo)metric,

2. the type $U[a,b][c,d]$ of such functions equipped with modulus of uniform continuity, on which we have a pseudometric, and

3. the type $UL[a,b][c,d]$ of such uniformly continuous functions additionally equipped with the structure of lifting to locators, which is not used for computing distances.

There are projection maps going up this list, which takes two elements with distance 0 in $U[a,b][c,d]$ into identical elements of $[a,b] \to [c,d]$.

Lemma 8.2.10. *For reals $a < b$ and $c < d$, the pseudometric space $UL[a,b][c,d]$ is Cauchy complete.*

Proof. Let $f_i : [a,b] \to [c,d]$, for $i : \mathbb{N}$, be elements of a Cauchy sequence of functions with corresponding moduli of uniform continuity $\omega_i : \mathbb{Q}_+ \to \mathbb{Q}_+$. Since the sequence is Cauchy, it is Cauchy pointwise, so that we can compute the underlying function g of the limit on the reals pointwise. This limit function g is uniformly continuous: for a given ε, we pick $N : \mathbb{N}$ such that $\rho(f_N, g) \leq \varepsilon/3$ and set $\omega(\varepsilon) :\equiv \omega_N(\varepsilon/3)$. The function g lifts to locators because if a sequence of reals comes equipped with locators for each element, then its limit can be assigned a locator, as in Lemma 6.8.1. □

8.3 Banach fixed point theorem

We will compute solutions to differential equations as fixpoints of a certain Lipschitz continuous endomap on a function space. In the Section 8.1, we discussed that in pseudometric spaces, limits are not unique, and hence our theorems will often conclude a structure rather than a property. For us, this means that, when developing precise statements of fixpoints of Lipschitz endomaps, we need to pay attention to when we have certain data, and when we merely know it to exist, as well as when continuous data provided by real numbers suffices, versus when we need to have the discrete data of rationals.

The development of the Banach fixed point theorem and the Picard-Lindelöf theorem in Section 8.4 is based on Royden and Fitzpatrick [83].

Definition 8.3.1. Given two pseudometric spaces (X, ρ) and (Y, σ), and a map $f : X \to Y$, a *Lipschitz constant* for f is a number $c : \mathbb{R}$ such that

$$(\forall x, y : X)\sigma(f(x), f(y)) \leq c\rho(x, y).$$

Lemma 8.3.2. *Let (X, ρ) and (Y, σ) be pseudometric spaces, and $f : X \to Y$ have a* rational *Lipschitz constant $c : \mathbb{Q}_+$. Then $\omega(\varepsilon) = \varepsilon/c$ is a modulus of uniform continuity for f.*

Corollary 8.3.3. *Let (X, ρ) and (Y, σ) be pseudometric spaces, and $f : X \to Y$ have a* real

Lipschitz constant $c : \mathbb{R}_+$. Then there exists *a modulus of uniform continuity for f.*

Proof. There exists $r : \mathbb{Q}$ with $c < r$, and so r is a *rational* Lipschitz constant for f. □

Lemma 8.3.4. *Let (X, ρ) be a* pseudometric *space, and $T : X \to X$ an endomap with Lipschitz constant $c : \mathbb{R}$. If $c < 1$ then a pseudo-fixed point of T is pseudo-unique, that is, if $\rho(T(x), x) = 0$ then for any $y : X$ with $\rho(T(y), y) = 0$ we have $\rho(x, y) = 0$.*

Proof. Suppose $\rho(T(x), x) = 0$ and $\rho(T(y), y) = 0$. Then

$$\rho(x, y) \le \rho(x, T(x)) + \rho(T(x), T(y)) + \rho(T(y), y) = c\rho(x, y).$$

If $\rho(x, y) > 0$ then this implies $\rho(x, y) < \rho(x, y)$, a contradiction. This argument shows that $\rho(x, y) \le 0$, hence $\rho(x, y) = 0$. □

Corollary 8.3.5. *Let (X, ρ) be a* metric *space, and $T : X \to X$ an endomap with Lipschitz constant $c : \mathbb{R}$. If $c < 1$, then the type of fixed points of T is a proposition.*

Proof. By Lemma 2.6.20, it suffices to prove that any two fixed points of T are equal. If x and y are fixed points of T, that is $T(x) = x$ and $T(y) = y$, then in particular $\rho(T(x), x) = \rho(T(y), y) = 0$, and so by Lemma 8.3.4 we get $\rho(x, y) = 0$, so $x = y$. □

Remark 8.3.6. In fact, in the above corollary it suffices to know that there *exists* a Lipschitz constant for f, as we are now showing a proposition, in contrast with Lemma 8.3.4.

For the remainder of this section, fix a pseudometric space (X, ρ) with a point $x_0 : X$, and an endomap $T : X \to X$. We aim to show that if T is Lipschitz, then we have a modulus of Cauchy convergence for the sequence $\lambda n.T^n(x_0)$. To phrase this, we define a logarithm-like function $\mathfrak{log}_c : \mathbb{Q}_+ \to \mathbb{N}$.

Lemma 8.3.7. *For $c : \mathbb{Q}$ with $0 < c < 1$ we can define $\mathfrak{log}_c : \mathbb{Q}_+ \to \mathbb{N}$ satisfying, for any $q : \mathbb{Q}_+$, that*

$$c^{\mathfrak{log}_c(q)} < q.$$

Proof. Find $d : \mathbb{N}$ with $\frac{1}{d} \leq q$, and take $a, b : \mathbb{N}$ coprime such that $c = \frac{a}{b}$, noting that $a > 0$ because $c > 0$. Since $c < 1$ we have $b = a + k$ with $k \geq 1$. We aim to find $n = \mathfrak{log}_c(q)$ such that

$$\left(\frac{a}{a+k}\right)^n < \frac{1}{d},$$

or equivalently

$$da^n < (a+k)^n.$$

By the binomial theorem, this is satisfied when

$$da^n < a^n + k^n + nka^{n-1},$$

or equivalently,

$$0 < a^{n-1}(a + nk - da) + k^n,$$

which certainly holds when we set $n = \mathfrak{log}_c(q) := da$. □

Lemma 8.3.8. *Suppose $T : X \to X$ has* rational *Lipschitz constant $0 < c < 1$. Suppose we have a* rational *$C : \mathbb{Q}$ with $\rho(T(x_0), x_0) \leq C$.*

The sequence $\lambda n.T^n(x_0)$ is Cauchy with modulus

$$M(\varepsilon) := \mathfrak{log}_c\left(\frac{\varepsilon(1-c)}{C}\right).$$

Proof. By induction we can show that for $k : \mathbb{N}$

$$\rho(x_{k+1}, x_k) \leq c^k \rho(x_1, x_0) \leq Cc^k.$$

Hence, if $m > k$, then, using the triangle inequality and the geometric sum formula,

$$\rho(x_m, x_k) \leq \sum_{i=k}^{m-1} \rho(x_{i+1}, x_i) \leq \sum_{i=k}^{m-1} Cc^i \leq C\frac{c^k - c^m}{1-c}.$$

Since $0 < c < 1$,

$$\rho(x_m, x_k) \leq \frac{Cc^k}{1-c},$$

which is strictly bounded from above by ε when

$$k \geq \mathfrak{log}_c\left(\frac{\varepsilon(1-c)}{C}\right).$$ □

Remark 8.3.9. Rather than the condition $\rho(T(x_0), x_0) \leq C$, we normally assume more generally that the space (X, ρ) itself has a *bound*, namely a number C so that

$$(\forall x, y : X)\rho(x, y) \leq C.$$

Remark 8.3.10. We use the stronger assumption that the Lipschitz constant c is a rational because Cauchy moduli are phrased in terms of rationals. In the more general case that we have a *real* Lipschitz constant $c : \mathbb{R}$ with $0 < c < 1$, we know that $(\exists q : \mathbb{Q})c < q < 1$, and hence we get the existence of a Cauchy modulus. Similarly, if we have a *real* bound $C : \mathbb{R}$ of the space (X, ρ), we get the existence of a Cauchy modulus.

As we will be dealing with real numbers equipped with locators in the next section, we may also assume the bound of X and Lipschitz constant of T to be a real with a locator, rather than the stronger requirement of them being rational numbers: after all, if we have a real with a locator, we can use Lemma 6.7.3 to compute appropriate rationals so that Lemma 8.3.8 applies.

In conclusion:

Theorem 8.3.11. *Let (X, ρ) be a pseudo-metric space with a point x_0. Suppose X is Cauchy complete and bounded by $C : \mathbb{Q}$. Let $T : X \to X$ be an endomap with a* rational *Lipschitz constant $0 < c < 1$.*

Then T has a pseudo-unique pseudo-fixed point, i.e. we have a point x with $\rho(T(x), x) = 0$, and for any $y : X$ with $\rho(T(y), y) = 0$ we have $\rho(x, y) = 0$.

Proof. Lemma 8.3.8 gives us a modulus for the sequence $\lambda n.T^n(x_0)$, so that by Cauchy completeness we can construct a limit x_∞. By Lemma 8.3.2, T is uniformly continuous, and hence preserves limits, so that $T(x_\infty)$ is a limit of $\lambda n.T^{n+1}(x_0)$. Now $\rho(T(x_\infty), x_\infty) = 0$ because they are limits of sequences which differ only by some offset. □

In the next section, we will use the above theorem to compute solutions of differential equations, where the functions involved lifts to locators, so that we get discrete data out.

8.4 Picard-Lindelöf

We will consider differential equations specified by a map $g : [x_0 - a, x_0 + a] \times [y_0 - b, y_0 + b] \to \mathbb{R}$ that takes an element of a product space. So we now give the pseudometric of such product spaces. After defining when g lifts to locators, we are ready to use the previous sections to compute a solution to differential equations in Theorem 8.4.7.

Definition 8.4.1. If $(X_1, \rho_1), \dots, (X_n, \rho_n)$ are pseudometric spaces, then the product $\prod_i X_i = X_1 \times \cdots \times X_n$ is given the pseudometric

$$\rho((x_1, \dots, x_n), (y_1, \dots, y_n)) := \sum_{i=1}^{n} \rho_i(x_i, y_i).$$

Additionally, if all ρ_i are metrics, then so is ρ.

Proof. We can show conditions 1–3 of Definition 8.1.1 for ρ when they do for all ρ_i using basic algebra. If all ρ_i satisfy condition 4, then we can show it for ρ using the fact that if a finite sum of nonnegative reals is 0, then all the summands must be 0. □

> *Remark* 8.4.2. This means that $\mathbb{R}^2$ gets assigned the taxicab metric on $\mathbb{R}^2$, rather than the Euclidean metric. This is just a matter of convenience.

Lemma 8.4.3. *Let (X_i, ρ_i) and (Y_i, σ_i) be pseudometric spaces, with i between 1 and n. Given $f_i : X_i \to Y_i$ for $i \in \{1, \dots, n\}$ with respective moduli of uniform continuity ω_i, the map $f_1 \times \cdots \times f_n : \prod_i X_i \to \prod_i Y_i$ is uniformly continuous with modulus*

$$\omega(\varepsilon) := \min_{i \in \{1,\dots,n\}} \omega_i(\varepsilon/n).$$

Proof. Let $x_i, x_i' : X_i$ for i between 1 and n be the components of two elements of the input space, and assume $\sum_{i=1}^n \rho_i(x_i, x_i') < \min_{i\in\{1,\dots,n\}} \omega_i(\varepsilon/n)$. In particular, this yields $\rho_i(x_i, x_i') < \omega_i(\varepsilon/n)$ for i between 1 and n, and so $\sigma_i(f_i(x_i), f_i(x_i')) < \varepsilon/n$, and hence $\sum_{i=1}^n \sigma_i(f_i(x_i), f_i(x_i')) < \varepsilon$, as required. □

Lemma 8.4.4. *The diagonal maps $X \to \prod_{i=1}^n X$ are uniformly continuous with modulus $\omega(\varepsilon) := \varepsilon/n$.*

Proof. If $x, x' : X$ with $\rho(x, x') < \varepsilon/n$, then $\sum_{i=1}^{n} \rho(x, x') < \varepsilon$. □

Corollary 8.4.5. *Let (X, ρ), (Y, σ) and (Z, τ) be pseudometric spaces. If $f : X \to Y$ is uniformly continuous with modulus ω and $g : X \times Y \to Z$ is uniformly continuous with modulus ψ, then $h(x) := g(x, f(x))$ is uniformly continuous with modulus $\xi(\varepsilon) := \min(\psi(\varepsilon)/2, \omega(\psi(\varepsilon)/2))/2$.*

Proof. The map h is the composition of a diagonal map $X \to X \times X$, composed with $\mathrm{id} \times f : X \times X \to X \times Y$, and finally $g : X \times Y \to Z$, so that the moduli of uniform continuity compose in reverse. □

Additionally, since the aim is to calculate a solution to a differential equation which lifts to locators in the sense of Definition 8.2.4, we need to define what it means for such a g to lift to locators. The following notion suffices as an abstract notion of computation.

Definition 8.4.6. With $a, b, c, d : \mathbb{R}$ and $a < b$, $c < d$, a function $g : [a, b] \times [c, d] \to \mathbb{R}$ *lifts to locators* if it comes equipped with a method for constructing a locator for $g(x, y)$, given both a locator for x and a locator for y. This means that g lifts to locators if it is equipped with an element of the type

$$(\Pi x : [a, b])(\Pi y : [c, d])\,\mathrm{locator}(x) \to \mathrm{locator}(y) \to \mathrm{locator}(g(x, y)).$$

Another way to say this is that g lifts to locators iff we can find the top edge L in the diagram

$$\begin{tikzcd}
{[a, b]^{\mathfrak{L}} \times [c, d]^{\mathfrak{L}}} \arrow[r] \arrow[d, "\mathrm{pr}_1 \times \mathrm{pr}_1"] \arrow[dr, phantom, "\circ"] & \mathbb{R}^{\mathfrak{L}} \arrow[d, "\mathrm{pr}_1"] \\
{[a, b] \times [c, d]} \arrow[r, "g"] & \mathbb{R}
\end{tikzcd}$$

where $[a, b]^{\mathfrak{L}} := (\Sigma x : [a, b])\,\mathrm{locator}(x)$ is the type of real numbers in the interval $[a, b]$ equipped with locators.

Theorem 8.4.7. *Let $x_0, y_0 : \mathbb{R}$ and $a, b : \mathbb{R}_+$ come equipped with locators. Suppose the function $g : [x_0 - a, x_0 + a] \times [y_0 - b, y_0 + b] \to \mathbb{R}$ has a modulus of uniform continuity, lifts to locators in the sense of Definition 8.4.6, and satisfies the following Lipschitz condition: we have a real $L : \mathbb{R}_+$*

equipped with a locator such that for all $x : [x_0 - a, x_0 + a]$ *and all* $y_1, y_2 : [y_0 - b, y_0 + b]$*, we have*

$$|g(x, y_1) - g(x, y_2)| \leq L\,|y_1 - y_2|\,.$$

Then we can compute a real l *with a locator, with* $0 < l \leq a$*, such that on* $[x_0 - l, x_0 + l]$ *we have a solution for differential equation*

$$f'(x) = g(x, f(x))$$

$$f(x_0) = y_0$$

in the sense that we can compute a uniformly continuous $f : [x_0 - l, x_0 + l] \to \mathbb{R}$ *that lifts to locators, with for all* $x : [x_0 - l, x_0 + l]$

$$f(x) = y_0 + \int_{x_0}^{x} g(x, f(x))\,\mathrm{d}x.$$

Proof. We first compute a number K such that $|g(x, y)| \leq K$ for all $x : [x_0 - a, x_0 + a]$ and $y : [y_0 - b, y_0 + b]$. We do this by applying the supremum construction of Lemma 8.2.2 pointwise to every function $|g(x, \cdot)| : [y_0 - b, y_0 + b] \to \mathbb{R}$ for each $x : [x_0 - a, x_0 + a]$, noting that these functions are indeed all uniformly continuous. Then Lemma 8.2.2 can be applied to the pointwise supremum to obtain K.

Conform to Royden and Fitzpatrick [83], we set

$$l \coloneqq \min(b/K, 1/2L).$$

We calculate the solution using Theorem 8.3.11 in the space $UL[x_0 - l, x_0 + l][y_0 - b, y_0 + b]$. We can use Lemma 6.6.1 to obtain a rational bound C of this space. The space has a point given by the constant function at y_0. It remains to construct an endomap on this space with a rational Lipschitz constant.

To construct the endomap on the space, for a given $f : UL[x_0 - l, x_0 + l][y_0 - b, y_0 + b]$ we compute the underlying function of $T(f)$ as in Royden and Fitzpatrick [83],

$$\lambda x. y_0 + \int_{x_0}^{x} g(t, f(t))\,\mathrm{d}t,$$

where the integrand is uniformly continuous by Corollary 8.4.5, and lifts to locators, so that by a version of Theorem 7.2.1, the integral has a locator whenever x does, and so the above function can be made into an element $T(f)$ of $UL[x_0 - l, x_0 + l][y_0 - b, y_0 + b]$.

The endomap T has a real Lipschitz constant lL: for arbitrary x we have

$$\begin{aligned}
|T(f)(x) - T(f')(x)| &= \left| \int_{x_0}^{x} [g(t, f(t)) - g(t, f'(t))] \,\mathrm{d}t \right| \\
&\leq \int_{x_0}^{x} |g(t, f(t)) - g(t, f'(t))| \,\mathrm{d}t \\
&\leq \int_{x_0}^{x} L\rho(f, f') \,\mathrm{d}t \\
&\leq |x - x_0| \, L\rho(f, f') \\
&\leq lL\rho(f, f'),
\end{aligned}$$

so that we get $\rho(T(f), T(f')) \leq lL\rho(f, f')$. Note that lL has a locator. By definition of l, we have $lL \leq 1/2 < 1$, so that by Lemma 6.7.3 we can find a rational Lipschitz constant $c < 1$ for T.

Hence by Theorem 8.3.11, T has a pseudo-unique pseudo-fixpoint with an underlying function f. Since f is a pseudo-fixpoint of T, by Lemma 8.2.9 the function $T(f)$ *equals* f, so that the desired equality holds. □

8.5 Notes

The notion of functions that *lift to locators* is used as an abstract model of computation. As discussed in Chapter 7, this notion is neither stronger nor weaker than continuity.

Picard iteration of an old endomap on a new pseudometric space gives a solution of a differential equation that computes, in the sense that when the resulting function is evaluated on reals with locators, then the output is a real with locator. In this way we can output, for instance, signed-digit representations of solutions of differential equations. The traditional proof was modified in three ways:

1. it was rephrased from set theory to type theory,

2. some assumptions are taken as structure rather than property, such as the data of a modulus of uniform continuity, and

3. we have assumed that the real numbers x_0, y_0, a, b and c involved come equipped with locators, to make the computation work.

The result in Theorem 8.4.7 tells us how we can type-theoretically compute differential equations with a pre-defined type of real numbers. This means that if we have certain initial data and parameters, not only is there a unique solution, but it can also be computed.

We have not found a definition of locators for general metric spaces, of which the locators introduced in Chapter 6 would be a special case. This would be an important future direction of research.

Another natural possibility to compute in metric spaces is to have the (pseudo)metric take values in a type $\mathbb{R}^{\mathfrak{L}}_{\geq 0}$ of nonnegative reals equipped with locators. We have not investigated this and leave it as a direction of future research. For us, the assumption that specific bounds and constants are equipped with locators suffices to compute.

Chapter 9

COMPUTATION IN PROOF ASSISTANTS

We briefly discuss how the mathematical ideas contained in this thesis may be formalized.

9.1 From inference rules to proof assistants

From a valid type-theoretic conclusion $\Gamma \vdash M : A$ we can systematically recover the derivation, namely the tree of inference rules that led to that conclusion, as we hinted to at the end of Section 2.2. This reconstruction, known as *type checking*, is done by various computer programs known as *proof assistants*. Most proof assistants do not output the derivation tree directly, but instead provide information *about* that tree, such as whether it exists, or, if there is a mistake in the conclusion, what part of the conclusion is problematic.

Another algorithm implemented in the vast majority of proof assistants is that of *normalization*, where a given term is simplified by use of judgmental equalities, introduced in Discussion 2.3.4. This is possible because judgmental equalities have a natural direction: for example, the computation rule

$$\Gamma \vdash \mathsf{ind}_{\mathbb{N}}(\lambda(x : \mathbb{N}).C, c_0, \lambda(x : \mathbb{N}).\lambda(y : C).c_s, 0) \equiv c_0 : C[0/x]$$

can be read as saying that $\mathsf{ind}_{\mathbb{N}}$ evaluated at input 0 should *simplify* to the term c_0. Note that the simplified term c_0 type-checks because the original term in terms of $\mathsf{ind}_{\mathbb{N}}$ did. There are a number of strategies for normalization. It is also possible to use hardware acceleration.

The relevance of this software ecosystem for us is that it makes type theory into a programming language, with type checking playing the role of compiling, and term normalization playing the role of program execution. One notable exception to the adequacy of current proof assistants is given by Brunerie's proof that $\pi_p(\mathbb{S}^3) = \mathbb{Z}/n\mathbb{Z}$, where $n : \mathbb{N}$ is a term that should normalize to 2: no proof assistant has yet been able to normalize n in realistic memory and time constraints [30, 4].

Regardless, proof assistants have been implemented that support UTT, including univalence and certain higher-inductive types such as propositional truncation [34, 96]. Such *cubical type theories* [32] can, at least in theory, reduce any term $n : \mathbb{N}$ to a numeral, so that they satisfy a *canonicity* property [54]. This makes UTT into a programming language, allowing us to compute with mathematical proofs, in a sense rather distinct from Russian constructivism [28]. There exist several libraries of formalized results from UTT [98, 15, 31, 66, 33, 6], implemented in a variety of proof assistants such as Coq, Agda, Cubical Agda [96], cubicaltt, Lean and RedPRL. There are also proof assistants that aim to implement a number of type theories and normalization strategies [14].

Our work deals with real numbers. A Dedekind real, say, can be formalized as a certain tuple whose main data is specified by *functions*, namely a predicate $\mathbb{Q} \to \mathrm{HProp}$. The canonical form of such a real does not tell us anything about its actual value, and this is one thing that can be addressed using our work. For instance, we can compute signed-digit representations without violating extensionality by formalizing Theorem 6.9.5, which says that locators and signed-digit representations are interdefinable.

9.2 Locators

In Chapter 6 we developed the theory of locators. To implement locators in a proof assistant, we need to make a choice about the type $\mathbb{R}$ of real numbers and the type $\mathbb{Q}$ of rationals.

1. These types of numbers can be defined explicitly as in Chapter 4. This will immediately allow us to compute. However, in this way we fix our choice of $\mathbb{R}$, which reduces modularity of the implementation.

2. The types may be introduced as axioms, so that the entire development is conditional on such a choice being made after the fact, or on the proof assistant being able to compute despite no explicit construction being used.

3. The entire development in this chapter may be *parameterized* by these types of numbers, so that specific constructions as in e.g. Chapter 4 can later be applied. This is the approach taken by Spitters and van der Weegen [89] and Gilbert [48].

Having somehow obtained appropriate types of numbers and operations on them, locators can be phrased in terms of them. For instance, in the Coq proof assistant, we may define locators as:

```
Definition locator (x : R) :=
    forall q r : Q, q < r -> ((i q) < x) + (x < (i r)).
```

where `i` is the inclusion of `Q` into `R` obtained from Lemma 4.3.3. Using *implicit coercions*, it may be possible to avoid using `i` explicitly.

Having formalized the notion of locators, we can pass around `Record`s consisting of a real with a locator for it. Another option may be to use the typeclass mechanism via `Class`es and `Instance`s, so that locators are constructed implicitly. The latter option allows us to write theorem statements and proofs that seem even closer to the original ones from plain constructive mathematics.

The equivalent presentation of locators established by Lemma 6.4.2 allows us to formalize locators in a perhaps more elementary way. Namely, recalling that DHProp $\simeq \mathbf{2}$, we can define a locator for $x : \mathbb{R}$ as a map $(\Pi q, r : \mathbb{Q}) q < r \longrightarrow \mathbf{2}$ satisfying certain correctness criteria. If desired, we can do away with the dependent product in the structural part of the formalization of locators by replacing our interval $q < r$ by an arbitrary $q : \mathbb{Q}$ and a *positive* rational $\varepsilon : \mathbb{Q}_{+}$, so that we may define a locators as:

```
Record locator (x : R) :=
  { ell : Q -> Q+ -> Bool
  ; ax1 : forall q : Q, forall e : Q+,
                ell q e = true  -> i q < x
  ; ax2 : forall q : Q, forall e : Q+,
                ell q e = false -> x < i (q + ' e)
  }.
```

where, as before, `i` includes the rationals into the reals, and additionally `'` includes the positive rationals into the rationals.

9.3 Metric spaces

The pseudometric space $UL[a,b][c,d]$ defined in Section 8.2 may be formalized in a proof assistant such as Coq, for instance as a `Record` consisting of the underlying function $[a,b] \to [c,d]$, a modulus of continuity, and the structure of lifting to locators. The method used to formalize the notion of a map $[a,b] \to [c,d]$ may have a great influence on the effort required to formalize further results. For instance, we can directly interpret $[a,b]$ and $[c,d]$ as Σ-types following Lemma 2.6.3, and find an appropriate programming technique to identify elements of them with their underlying element of $\mathbb{R}$. We may also follow Definition 2.6.16 and see it as a map

$$(\Pi x : \mathbb{R}) x \in [a,b] \to (\Sigma y : \mathbb{R}) y \in [c,d],$$

which requires a nontrivial definition of function composition and other basic notions.

As in the previous section, it may be beneficial to store the modulus of continuity and the structure of lifting to locators using typeclasses, so that we can simply pass around the underlying function without explicitly choosing specific moduli of continuity or structures of lifting to locators.

In order to compute solutions of differential equations following Theorem 8.4.7, we can similarly use a certain `Record` type whose elements are functions equipped with moduli of uniform continuity and the structure of lifting to locators, and define the endomap T on that type. Part of the proof obligation here will be to prove that the underlying function of $T(f)$ is indeed a function whose output values range between $y_0 - b$ and $y_0 + b$, that it is uniformly continuous, and that it lifts to locators, as we have sketched in the proof of Theorem 8.4.7.

9.4 Haskell prototype

Having recalled inference rules as introduced in Chapter 2, we discussed how proof assistants are able to compute with mathematical proofs directly. We have given some initial thoughts and definitions towards a development of the work in this thesis in a proof assistant, which would allow executing the algorithms contained in the mathematical proofs.

In order to test the feasibility of such an implementation, we have implemented the mathematical ideas in Haskell [25]. Although Haskell does not have the expressivity of dependent type theory, it still gives some idea about feasibility, as proof assistants such as Coq and Agda can compile dependent type theory down to Haskell, so that a naive Haskell implementation should be at least as good as a naive Coq or Agda implementation.

The type of reals equipped with locators is represented as the following type `RL`.

```
data Location = LocatesLeft | LocatesRight
type RL = Rational -> Rational -> Location
```

The locator accepts two rationals `q` and `r` as input, where it is a requirement that `q<r`, although this is not checked by the Haskell code. For a locator `l` evaluated at `q` and `r`, an output value of `LocatesLeft` indicates that the real is on the left of `r`, and an output value of `LocatesRight` indicates that the real is on the right of `q`.

Compared to the definition in UTT, we have dropped the data specifying the real itself, keeping only the locator. This is justified because given only the locator, we can recover the corresponding subsets of the rationals, and correspondingly phrase when the data of a locator specifies a Dedekind cut. Since we have locators, it may be possible to capture the Dedekind cut axioms in Haskell using Theorem 6.10.3, although our implementation does not include this.

Having defined locators, we can work towards the terminology of Lemma 6.4.2.

```
locatesLeft :: RL -> Rational -> Rational -> Bool
locatesLeft l q r = l q r == LocatesLeft
locatesRight :: RL -> Rational -> Rational -> Bool
locatesRight l q r = l q r == LocatesRight
```

`locatesLeft l q r` is true when the real locates to the left of `r`. That is, if we think of `l` as representing a real `x`, then `locatesLeft l q r -> (x < r)`. However, this property is not enforced by Haskell.

A naive way to compute lower bounds is to find the first element in a list of decreasing integers that locates right:

```
-- Compute a rational lower bound of x
lowerBound :: RL -> Rational
lowerBound l = fromInteger $ head
    [ q-1 | q <- [0,-1..]
          , locatesRight l (fromInteger (q-1)) (fromInteger q)
          ]
```

Many constructions are implemented in various ways. For instance, lower bounds can also be computed using a method which only tries the negations of powers of two, rather than all negative integers.

This Haskell implementation shows that the essence of the ideas contained in this thesis works, in the sense that simple calculations with locators work, including the constructions in Sections 6.3–6.9. For instance, the locators for $-x$, $\max(x, y)$ and $|x|$ can be constructed as follows.

```
-- Construct a locator for -x from a locator for x
mkMinus :: RL -> RL
mkMinus l q r = case l (-r) (-q) of -- Note: if q < r then -r < -q
  LocatesLeft  -> LocatesRight -- if x < -q then q < -x
  LocatesRight -> LocatesLeft  -- if -r < x then -x < r
```

```
-- Construct a locator for max(x, y)
mkMax :: RL -> RL -> RL
mkMax l m q r =
  case (l q r, m q r) of
    (LocatesLeft, LocatesLeft) -> LocatesLeft -- x < r and y < r
    _                          -> LocatesRight

-- Locator for |x|
mkAbs :: RL -> RL
mkAbs l = mkMax l (mkMinus l)
```

A particular shortcoming, which is common for exact real algebra, occurs with deeply nested algebraic combinations, such as when $2 + (2 + \ldots (2 + 2) \ldots)$ is computed by using the locator for the rational 2 followed by successive construction of locators for addition as in Theorem 6.7.6: locators constructed in this way are often not usable in practice. The construction of locators for multiplication is particularly troublesome: computation of a 5-digit signed digit representation of $2 \cdot (2 \cdot (2 \cdot (2 \cdot 2)))$ takes 23 seconds on an Intel Core i5-5200U processor.

Chapter 10

CLOSING REMARKS

Univalent type theory (UTT) allows to define types of reals whose identity types directly capture the intended equality of real numbers. We have defined the required structure of real numbers, namely that of a Cauchy complete Archimedean ordered field, and carried out several specific constructions of types of numbers, in Chapter 4. Some types of reals can be conveniently characterized by a universal property, which allowed us to make certain new relations between such specific constructions in Chapter 5.

As a future direction, it should be the case that a uniformly continuous map $\mathbb{Q}^n \to \mathbb{R}$ can be extended to a map $\mathbb{R}^n \to \mathbb{R}$.

Mathematical theorems are proven in UTT by writing *terms* of a *type*. As we have discussed in our mentioning of the inference rules of type theory in Chapter 2, this means that, if we are only interested in proving theorems, we do not need to concern ourselves with the proof tree, as this can be recovered from the final term. *Proof assistants* execute this reconstruction in a process called *proof checking*, and additionally *normalize* terms.

The latter process of normalization leads us to *computation* in type theory: the mathematical proofs can be seen as algorithms which we can execute. This makes most sense for terms of discrete types such as the Booleans and disjoint unions since, for instance, the normal form of functions are λ terms with little interpretability.

Computation with real numbers is well understood in intensional systems such as in a formalization based on setoids [36, 64, 21, 22]. Computation with real numbers has also been studied from a more programming-theoretic point of view, such as in Di Gianantonio [37] and Escardó [42].

In UTT we are hindered from following the setoidal approach of making observations of real numbers using *operators* that violate the equivalence condition, because every term in type theory respects identity types.

In order to solve this, we have paid attention to the difference between property and structure while defining the real numbers and other foundations of constructive analysis. We have equipped the reals with a natural structure to observe information of real numbers, such as signed-digit expansions.

The constructions and results remind of computable analysis. But our development is orthogonal to computability: even reals that are not computable in some semantics can have locators, for example in the presence of excluded middle, in which case all reals have locators. And our abstract notion of computation, given by functions that *lift to locators*, does not coincide with continuity, in contrast with computable analysis. Specifically, if all continuous functions automatically lift to locators, we obtain WLPO, and conversely using PEM we can construct a discontinuous function that lifts to locators, so that lifting to locators is neither weaker nor stronger than continuity.

We have shown how to find Cauchy sequences from our locators, and we can similarly obtain a sequence of nested intervals for a real with a locator.

Our work allows to obtain signed-digit representations of integrals. These results are based on backwards error propagation, essentially due to our notion of lifting to locators. The advantage of this is that we are guaranteed to be able to find results. However, forward error propagation, as in Martin-Dorel and Melquiond [74] and Mahboubi, Melquiond, and Sibut-Pinote [73], may be more efficient. It may be possible to combine the naturalness of locators with forward error propagation by equipping the real numbers involved with bounds as in Remark 6.6.2. Having shown that we can compute arbitrarily precise approximations to reals with locators in Lemma 6.6.3, we may as well equip real numbers with an efficient method for doing so. Thus, in future work, some of the techniques of previous work on verified computation with exact reals may be developed in our setting as well.

We have shown a new exact intermediate value theorem in Section 7.3, showing that locators can also be useful in their own right when computation is not the goal, although the root we construct does come equipped with a locator and hence can be computed.

In Chapters 6–8 we have assumed a Cauchy complete Archimedean ordered field $\mathbb{R}$. However, as we showed in Theorem 6.9.7, the only numbers equipped with locators we encounter are elements of the *Cauchy reals*, a type which is not an example of a Cauchy complete Archimedean ordered field. This means that the requirement of Cauchy completeness is too strong: we are asking for a larger type of reals than we actually end up using.

We solved ordinary differential equations as uniformly continuous functions that lift to locators. This solution is developed in pseudometric spaces, so that we can *equip* the solution with a modulus of uniform continuity and the structure of lifting to locators, rather than this data merely existing. The proof follows a traditional Picard iteration, showing that the use of locators is not a burden, and additionally allow to compute.

A possible future direction is to find a more general notion of locator that applies to more general spaces, such as the complex plane, function spaces, or (pseudo)metric spaces. A related question is that of the appropriate notion of locator for variations of Dedekind reals such as the extended reals.

Another possible direction of future research is to apply locators to other areas of analysis.

The work lends itself to being formalized in a proof assistant such as Agda or Coq, for the sake of automatically obtaining algorithms from proofs. In fact, this was included in our original aims, but it proved to be too ambitious for our time-limited project, although we did implement a Haskell prototype, discussed in Section 9.4. One particular issue that will somehow need to be dealt with is our widespread identification of elements of subtypes with their underlying element of the supertype, such as the identification of rationals with their embedding in the reals, and in the formalization of maps $[a, b] \to [c, d]$ of closed intervals. It would be more than convenient to have a proof-theoretic notion of subtyping that agreed with the univalent notion described in Section 2.6, justifying this identification.

We may worry that the proofs we provided are not sufficiently efficient for useful calculations, and we intend to address this important issue in future work. It is also possible that the chosen definition of the real numbers, as in Section 9.2, may impact computational efficiency, despite us actually being interested in the discrete data obtained from the locators rather than the underlying reals.

Index

Symbols

$\mathbb{N}$, 13

$\mathbb{Q}$, 76

$\mathbb{Q}_+$, 76

$\mathbb{R}_{\mathbf{C}}$, 80

$\mathbb{R}_{\mathbf{D}}$, 90

$\mathbb{R}_{\mathbf{E}}$, 100

$\mathbb{R}_{\mathbf{H}}$, 86

$\mathbb{R}^{\mathfrak{L}}$, 148

$\mathbb{Z}$, 75

$\mathcal{C}_R$, 82

$\mathcal{C}_Y$, 103

$\mathcal{U}$, 14

$\mathcal{U}_i$, 14

$\mathcal{P}$, 47

X/R, 55

CS-hom, 84

DHProp, 33

LPO, 57

PEM, 56

HProp, 29

WEM, 57

WLPO, 57

ap, 26

isCut§, 142

efq, 10

eq, 84

locatesRight, 125

locatesLeft, 125

id, 11

im, 34

ind$_{\mathbb{N}}$, 19–20

ind$_{=}$, 25–26

ind$_{X+Y}$, 24

inl, 17

inr, 17

isContr, 39

isCut, 91, 141

isHProp, 29

isSet, 38

$\lim_{n\to\infty}$, 79

lim, 84

$\mathfrak{log}_c$, 164

max, 72

min, 72

rat, 84

$\mathsf{rec}_{\mathbf{0}}$, 10

rec_{X+Y}, 10

refl, 18

sq, 30

sub-CS-hom, 104

succ, 13

0, 21

1, 21

$=$, 18–19

$u =_p^B v$, 87

$\equiv$, 20

Π, 15–16

$+$, 17

Σ, 18

$\times$, 28

$\land$, 32

$\lor$, 32

#, 70

$\forall$, 32

$\exists$, 32

$\Rightarrow$, 32

$\Leftrightarrow$, 32

$\neg$, 32

$\in$, 47

$\leq$, 62

$\cdot$, 73

$\neq$, 70

$\sim$, 81

$\top$, 31

$\bot$, 31

$\|\cdot\|$, 30–31

$|\cdot|$, 30

$\star$, 21

A

antisymmetric, 62

apartness relation, 70

Archimedean, 77

axiom of choice, 58

B

binary relation, 55

bounded, 89

C

Cauchy approximation, 82

Cauchy complete, 79

center of contraction, 39

constructive field, 70

continuous, 148

contractible, 39

convex body, 112

cotransitive, 62

countable choice, 58

D

decidable, 33

dependent identification, 87

dependent type theory, 7

disjoint, 90

E

embedding, 47

equivalence class, 55

F

function extensionality, 37

I

interval object, 115

irreflexive, 62

iteration property, 112

J

judgmental equality, 20

L

lattice, 72

lifting to locators, 147, 161, 168

limit, 78, 82, 83

limited principle of omniscience, 57

Lipschitz constant, 163

locally nonconstant, 152

located, 90

locator, 120

M

metric, 156

modulus of Cauchy convergence, 82

modulus of Cauchy convergence, 78

modulus of uniform continuity, 148, 158

N

naive independent function extensionality, 38

O

ordered field, 73

P

partial order, 62

pointwise continuous, 148

predicate, 35

premetric, 81

preorder, 62

preserving limits, 159

proposition, 29

pseudometric, 157

Q

quotient type, 55

R

reflexive, 62

rounded, 90

S

set, 38

strict partial order, 70

subset, 47

subtype, 47

symmetric, 62

T

transitive, 62, 90

truncation, 30

type family, 15

U

univalent type theory, 7

UTT, 7

W

weak limited principle of omniscience, 57

Bibliography

[1] B. Ahrens, C. Kapulkin, and M. Shulman. “Univalent categories and the Rezk completion”. In: *Mathematical Structures in Computer Science* 25.5 (Jan. 2015), pp. 1010–1039. ISSN: 1469-8072. DOI: 10.1017/s0960129514000486.

[2] B. Ahrens, P. Capriotti, and R. Spadotti. “Non-wellfounded trees in Homotopy Type Theory”. In: *ArXiv e-prints* (Apr. 2015). arXiv: 1504.02949 [cs.LO].

[3] T. Altenkirch, N. A. Danielsson, and N. Kraus. “Partiality, Revisited - The Partiality Monad as a Quotient Inductive-Inductive Type”. In: *Foundations of Software Science and Computation Structures - 20th International Conference, FOSSACS 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala, Sweden, April 22-29, 2017, Proceedings*. 2017, pp. 534–549. DOI: 10.1007/978-3-662-54458-7_31.

[4] C. Angiuli and R. Harper. “Meaning explanations at higher dimension”. In: *Indagationes Mathematicae* 29.1 (2018), pp. 135–149. ISSN: 0019-3577. DOI: 10.1016/j.indag.2017.07.010.

[5] C. Angiuli, K. Hou, and R. Harper. “Computational Higher Type Theory III: Univalent Universes and Exact Equality”. In: *CoRR* abs/1712.01800 (2017). arXiv: 1712.01800.

[6] C. Angiuli et al. “The RedPRL Proof Assistant (Invited Paper)”. In: *Proceedings of the 13th International Workshop on Logical Frameworks and Meta-Languages: Theory and Practice, LFMTP@FSCD 2018, Oxford, UK, 7th July 2018*. 2018, pp. 1–10. DOI: 10.4204/EPTCS.274.1.

[7] R. Atkey, N. Ghani, and P. Johann. “A relationally parametric model of dependent type theory”. In: *The 41st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL ’14, San Diego, CA, USA, January 20-21, 2014.* 2014, pp. 503–516. DOI: 10.1145/2535838.2535852.

[8] S. Awodey, J. Frey, and S. Speight. “Impredicative Encodings of (Higher) Inductive Types”. In: *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science.* LICS ’18. Oxford, United Kingdom: ACM, 2018, pp. 76–85. ISBN: 978-1-4503-5583-4. DOI: 10.1145/3209108.3209130.

[9] S. Awodey, N. Gambino, and K. Sojakova. “Homotopy-initial algebras in type theory”. In: *ArXiv e-prints* (Apr. 2015). arXiv: 1504.05531 [math.LO].

[10] S. Awodey, N. Gambino, and K. Sojakova. “Inductive Types in Homotopy Type Theory”. In: *Logic in Computer Science (LICS), 2012 27th Annual IEEE Symposium on.* June 2012, pp. 95–104. DOI: 10.1109/LICS.2012.21.

[11] S. Awodey and M. A. Warren. “Homotopy theoretic models of identity types”. In: *Math. Proc. Cambridge Philos. Soc.* 146.1 (2009), pp. 45–55. ISSN: 0305-0041. DOI: 10.1017/S0305004108001783. arXiv: 0709.0248.

[12] A. Bauer and other contributors. *A formalization of the Dedekind reals in Coq.* Accessed July 2019. URL: https://github.com/andrejbauer/dedekind-reals.

[13] A. Bauer and P. Taylor. “The Dedekind reals in abstract Stone duality”. In: *Math. Struct. Comput. Sci.* 19.4 (2009), pp. 757–838. DOI: 10.1017/S0960129509007695.

[14] A. Bauer et al. “Design and Implementation of the Andromeda Proof Assistant”. In: *CoRR* abs/1802.06217 (2018). arXiv: 1802.06217.

[15] A. Bauer et al. “The HoTT library: a formalization of homotopy type theory in Coq”. In: *Proceedings of the 6th ACM SIGPLAN Conference on Certified Programs and Proofs, CPP 2017, Paris, France, January 16-17, 2017.* 2017, pp. 164–172. DOI: 10.1145/3018610.3018615.

[16] J. Bernardy, P. Jansson, and R. Paterson. “Proofs for free: Parametricity for dependent types”. In: *J. Funct. Program.* 22.2 (2012), pp. 107–152. DOI: 10.1017/S0956796812000056.

[17] E. Bishop. “A general language”. URL: http://www.cs.bham.ac.uk/~mhe/Bishop/AGeneralLanguage.pdf.

[18] E. Bishop. *Foundations of constructive analysis.* Vol. 60. McGraw-Hill series in higher mathematics. New York: McGraw-Hill Book Co., 1967. ISBN: 4871877140.

[19] E. Bishop. “How to compile mathematics into Algol”. URL: http://www.cs.bham.ac.uk/~mhe/Bishop/Algol.pdf.

[20] E. Bishop and D. Bridges. *Constructive analysis.* Berlin New York: Springer-Verlag, 1985. ISBN: 978-3-642-64905-9. DOI: 10.1007/978-3-642-61667-9.

[21] S. Boldo, C. Lelay, and G. Melquiond. “Coquelicot: A User-Friendly Library of Real Analysis for Coq”. In: *Mathematics in Computer Science* 9.1 (2015), pp. 41–62. DOI: 10.1007/s11786-014-0181-1.

[22] S. Boldo, C. Lelay, and G. Melquiond. “Formalization of real analysis: a survey of proof assistants and libraries”. In: *Mathematical Structures in Computer Science* 26.7 (2016), pp. 1196–1233. DOI: 10.1017/S0960129514000437.

[23] A. B. Booij. “Extensional constructive real analysis via locators”. In: *ArXiv e-prints* (2018). arXiv: 1805.06781 [math.LO].

[24] A. B. Booij et al. “Parametricity, Automorphisms of the Universe, and Excluded Middle”. In: *22nd International Conference on Types for Proofs and Programs (TYPES 2016).* Ed. by S. Ghilezan, H. Geuvers, and J. Ivetić. Vol. 97. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018, 7:1–7:14. ISBN: 978-3-95977-065-1. DOI: 10.4230/LIPIcs.TYPES.2016.7.

[25] A. B. Booij. *Haskell implementation of reals equipped with locators.* Accessed January 2020. URL: https://github.com/abooij/haskell-locators.

[26] A. B. Booij. “The HoTT reals coincide with the Escardó-Simpson reals”. In: *CoRR* abs/1706.05956 (2017). arXiv: 1706.05956.

[27] N. Bourbaki. *Elements of mathematics: General Topology, part 2*. Actualités scientifiques et industrielles. Berlin: Springer, 1998. ISBN: 9783540645634.

[28] D. Bridges and F. Richman. *Varieties of Constructive Mathematics*. London Mathematical Society Lecture Notes 97. Cambridge University Press, 1987. ISBN: 0521318025.

[29] L. E. Brouwer. “Besitzt jede reelle Zahl eine Dezimalbruchentwicklung?” In: *Mathematische Annalen* 83 (1921), pp. 201–210. URL: http://eudml.org/doc/158869.

[30] G. Brunerie. “On the homotopy groups of spheres in homotopy type theory”. PhD thesis. Université de Nice, 2016. arXiv: 1606.05916 [math.AT].

[31] G. Brunerie et al. *Homotopy Type Theory in Agda*. Accessed February 2018. URL: https://github.com/HoTT/HoTT-Agda.

[32] E. Cavallo, A. Mörtberg, and A. W. Swan. “Unifying Cubical Models of Univalent Type Theory”. In: *28th EACSL Annual Conference on Computer Science Logic, CSL 2020, January 13-16, 2020, Barcelona, Spain*. Ed. by M. Fernández and A. Muscholl. Vol. 152. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, 14:1–14:17. DOI: 10.4230/LIPIcs.CSL.2020.14.

[33] C. Cohen et al. “Cubical Type Theory: A Constructive Interpretation of the Univalence Axiom”. In: *FLAP* 4.10 (Nov. 2017), pp. 3127–3170. URL: http://collegepublications.co.uk/ifcolog/?00019.

[34] 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 ’18. Oxford, United Kingdom: ACM, 2018, pp. 255–264. ISBN: 978-1-4503-5583-4. DOI: 10.1145/3209108.3209197.

[35] T. Coquand and A. Spiwack. “Towards Constructive Homological Algebra in Type Theory”. In: *Towards Mechanized Mathematical Assistants, 14th Symposium, Calcule-*

mus 2007, 6th International Conference, MKM 2007, Hagenberg, Austria, June 27-30, 2007, Proceedings. 2007, pp. 40–54. DOI: 10.1007/978-3-540-73086-6_4.

[36] L. Cruz-Filipe. "Constructive Real Analysis: a Type-Theoretical Formalization and Applications". PhD thesis. University of Nijmegen, Apr. 2004.

[37] P. Di Gianantonio. "A Functional Approach to Computability on Real Numbers". PhD thesis. Universita di Pisa-Genova-Udine, 1993.

[38] M. Escardó. *Introduction to Univalent Foundations of Mathematics with Agda: The powerset in the presence of propositional resizing*. Accessed October 2019. Oct. 19, 2019. URL: https://www.cs.bham.ac.uk/~mhe/HoTT-UF-in-Agda-Lecture-Notes/HoTT-UF-Agda.html#powerset-resizing.

[39] M. Escardó. "Categorical axioms for functional real-number computation". Accessed April 2020. Nov. 2011. URL: https://www.cs.bham.ac.uk/~mhe/.talks/map2011/parts-1-2-escardo.pdf.

[40] M. H. Escardó. *Message to the Univalent Foundations mailing list*. https://groups.google.com/d/msg/univalent-foundations/SA0dzenV1G4/d5iIGdKKNxMJ. Accessed July 2019. Mar. 2013.

[41] M. H. Escardó. *Message to the Univalent Foundations mailing list*. https://groups.google.com/d/msg/homotopytypetheory/-5mLEi_qMTo/GyTkmhIRK1AJ. Accessed July 2019. Mar. 2014.

[42] M. H. Escardó. "PCF extended with real numbers: a domain-theoretic approach to higher-order exact real number computation". PhD thesis. University of Edinburgh, 1997.

[43] M. H. Escardó and A. K. Simpson. "A Universal Characterization of the Closed Euclidean Interval". In: *16th Annual IEEE Symposium on Logic in Computer Science, Boston, Massachusetts, USA, June 16-19, 2001, Proceedings*. Draft full version at http://www.cs.bham.ac.uk/~mhe/papers/interval.pdf, accessed August 2019. 2001, pp. 115–125. DOI: 10.1109/LICS.2001.932488.

[44] M. H. Escardó and C. Xu. “The Inconsistency of a Brouwerian Continuity Principle with the Curry-Howard Interpretation”. In: *13th International Conference on Typed Lambda Calculi and Applications, TLCA 2015, July 1-3, 2015, Warsaw, Poland.* 2015, pp. 153–164. DOI: 10.4230/LIPIcs.TLCA.2015.153.

[45] M. Frank. “Interpolating Between Choices for the Approximate Intermediate Value Theorem”. In: *ArXiv e-prints* (Jan. 2017). arXiv: 1701.02227 [math.LO].

[46] N. Gambino and R. Garner. “The identity type weak factorisation system”. In: *Theoretical Computer Science* 409.1 (2008), pp. 94–109. ISSN: 0304-3975. DOI: https://doi.org/10.1016/j.tcs.2008.08.030.

[47] R. Garner. “On the strength of dependent products in the type theory of Martin-Löf”. In: *Annals of Pure and Applied Logic* 160.1 (July 2009), pp. 1–12. ISSN: 0168-0072. DOI: 10.1016/j.apal.2008.12.003.

[48] G. Gilbert. “Formalising real numbers in homotopy type theory”. In: *Proceedings of the 6th ACM SIGPLAN Conference on Certified Programs and Proofs, CPP 2017, Paris, France, January 16-17, 2017.* 2017, pp. 112–124. DOI: 10.1145/3018610.3018614.

[49] J.-Y. Girard. “Interprétation fonctionelle et élimination des coupures de l’arithmétique d’ordre supérieur”. PhD thesis. Université Paris VII, 1972.

[50] M. Hedberg. “A coherence theorem for Martin-Löf’s type theory”. In: *J. Functional Programming* (1998), pp. 4–8.

[51] M. Hendtlass. “The intermediate value theorem in constructive mathematics without choice”. In: *Ann. Pure Appl. Logic* 163.8 (2012), pp. 1050–1056. DOI: 10.1016/j.apal.2011.12.026.

[52] M. Hofmann. “Extensional concepts in intensional type theory”. PhD thesis. University of Edinburgh, 1995.

[53] M. Hofmann and T. Streicher. “The groupoid interpretation of type theory”. In: *Twenty-five years of constructive type theory (Venice, 1995).* Vol. 36. Oxford Logic Guides. New York: Oxford Univ. Press, 1998, pp. 83–111.

[54] S. Huber. “Canonicity for Cubical Type Theory”. In: *J. Autom. Reasoning* 63.2 (2019), pp. 173–210. DOI: 10.1007/s10817-018-9469-1.

[55] H. Ishihara. “Sequentially Continuity in Constructive Mathematics”. In: *Combinatorics, Computability and Logic.* Ed. by C. S. Calude, M. J. Dinneen, and S. Sburlan. London: Springer London, 2001, pp. 5–12. ISBN: 978-1-4471-0717-0. DOI: 10.1007/978-1-4471-0717-0_2.

[56] P. T. Johnstone. *Sketches of an elephant: a topos theory compendium.* Oxford New York: Oxford University Press, 2002. ISBN: 978-0198524960.

[57] T. de Jong. *The Scott model of PCF in univalent type theory.* 2019. arXiv: 1904.09810 [math.LO].

[58] C. Kapulkin and P. L. Lumsdaine. “The Simplicial Model of Univalent Foundations (after Voevodsky)”. In: *ArXiv e-prints* (Nov. 2012). arXiv: 1211.2851 [math.LO].

[59] C. Kapulkin, P. L. Lumsdaine, and V. Voevodsky. “Univalence in Simplicial Sets”. In: *ArXiv e-prints* (Mar. 2012). arXiv: 1203.2553 [math.AT].

[60] C. Knapp. “Partial functions and recursion in univalent type theoy”. PhD thesis. University of Birmingham, 2018.

[61] N. Kraus. *The truncation map* $|\_| : \mathbb{N} \to ||\mathbb{N}||$ *is nearly invertible.* Accessed September 2019. Oct. 28, 2013. URL: https://homotopytypetheory.org/2013/10/28/the-truncation-map-_-%E2%84%95-%E2%80%96%E2%84%95%E2%80%96-is-nearly-invertible/.

[62] N. Kraus et al. “Generalizations of Hedberg’s Theorem”. In: *Typed Lambda Calculi and Applications.* Ed. by M. Hasegawa. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 173–188. ISBN: 978-3-642-38946-7.

[63] N. Kraus et al. “Notions of Anonymous Existence in Martin-Löf Type Theory”. In: *Logical Methods in Computer Science* 13.1 (2017). DOI: 10.23638/LMCS-13(1:15)2017.

[64] R. Krebbers and B. Spitters. “Type classes for efficient exact real arithmetic in Coq”. In: *Logical Methods in Computer Science* 9.1:1 (2013), pp. 1–27. DOI: 10.2168/LMCS-9(1:01)2013.

[65] J. Lambek and P. Scott. *Introduction to Higher Order Categorical Logic*. Cambridge Studies in Advanced Mathematics 7. Cambridge University Press, 1986. ISBN: 0521246652.

[66] D. R. Licata and G. Brunerie. “A Cubical Approach to Synthetic Homotopy Theory”. In: *30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, July 6-10, 2015*. 2015, pp. 92–103. DOI: 10.1109/LICS.2015.19.

[67] D. R. Licata and R. Harper. “Canonicity for 2-dimensional type theory”. In: *Proceedings of the 39th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2012, Philadelphia, Pennsylvania, USA, January 22-28, 2012*. 2012, pp. 337–348. DOI: 10.1145/2103656.2103697.

[68] R. S. Lubarsky. “On the Cauchy Completeness of the Constructive Cauchy Reals”. In: *Electr. Notes Theor. Comput. Sci.* 167 (2007), pp. 225–254. DOI: 10.1016/j.entcs.2006.09.012.

[69] P. L. Lumsdaine. *Strong functional extensionality from weak*. Accessed August 2019. Dec. 19, 2011. URL: https://homotopytypetheory.org/2011/12/19/strong-funext-from-weak/.

[70] P. L. Lumsdaine and M. Shulman. “Semantics of higher inductive types”. In: *Mathematical Proceedings of the Cambridge Philosophical Society* (June 2019), pp. 1–50. DOI: 10.1017/S030500411900015X.

[71] S. Mac Lane. *Categories for the Working Mathematician*. Graduate Texts in Mathematics 5. Springer-Verlag, 1971. ISBN: 0387900357.

[72] S. Mac Lane and I. Moerdijk. *Sheaves in Geometry and Logic: a First Introduction to Topos Theory*. Universitext. Springer-Verlag, 1992. ISBN: 0387977104.

[73] A. Mahboubi, G. Melquiond, and T. Sibut-Pinote. “Formally Verified Approximations of Definite Integrals”. In: *Interactive Theorem Proving–7th International Conference, ITP 2016, Nancy, France, August 22-25, 2016, Proceedings.* 2016, pp. 274–289. DOI: 10.1007/978-3-319-43144-4_17.

[74] É. Martin-Dorel and G. Melquiond. “Proving Tight Bounds on Univariate Expressions with Elementary Functions in Coq”. In: *J. Autom. Reasoning* 57.3 (2016), pp. 187–217. DOI: 10.1007/s10817-015-9350-4.

[75] P. Martin-Löf. “An intuitionistic theory of types”. In: *Twenty-five years of constructive type theory (Venice, 1995).* Ed. by G. Sambin and J. M. Smith. Vol. 36. Oxford Logic Guides. Oxford University Press, 1998, pp. 127–172.

[76] B. Nordström, K. Petersson, and J. M. Smith. *Programming in Martin-Löf’s Type Theory: An Introduction.* Oxford University Press, July 1990. ISBN: 0198538146.

[77] R. O’Connor. “Incompleteness & Completeness: Formalizing Logic and Analysis in Type Theory”. PhD thesis. Radboud Universiteit Nijmegen, 2009.

[78] I. Orton and A. M. Pitts. “Axioms for Modelling Cubical Type Theory in a Topos”. In: *25th EACSL Annual Conference on Computer Science Logic, CSL 2016, August 29 - September 1, 2016, Marseille, France.* 2016, 24:1–24:19. DOI: 10.4230/LIPIcs.CSL.2016.24.

[79] D. Pataraia. “A constructive proof of the fixed-point theorem for dcpo’s”. Unpublished manuscript presented at the 65th Peripatetic Seminar on Sheaves and Logic, in Aarhus, Denmark, November 1997.

[80] F. Richman. “Real numbers and other completions”. In: *Mathematical Logic Quarterly* 54.1 (2008), pp. 98–108. ISSN: 1521-3870. DOI: 10.1002/malq.200710024.

[81] E. Rijke. *The join construction.* 2017. arXiv: 1701.07538 [math.CT].

[82] E. Rijke and B. Spitters. “Sets in homotopy type theory”. In: *Mathematical Structures in Computer Science* 25.5 (Jan. 2015), pp. 1172–1202. DOI: 10.1017/S0960129514000553.

[83] H. L. Royden and P. M. Fitzpatrick. *Real analysis*. Pearson, 2010. ISBN: 978-0-13-143747-0.

[84] P. Schuster. "Unique existence, approximate solutions, and countable choice". In: *Theor. Comput. Sci.* 305.1-3 (2003), pp. 433–455. DOI: 10.1016/S0304-3975(02)00707-7.

[85] P. Schuster. "What is Continuity, Constructively?" In: *J. UCS* 11.12 (2005), pp. 2076–2085. DOI: 10.3217/jucs-011-12-2076.

[86] P. Schuster and H. Schwichtenberg. "Constructive Solutions of Continuous Equations". In: *de Gruyter Series in Logic and Its Applications*. Ed. by G. Link. Walter de Gruyter, Jan. 2004. DOI: 10.1515/9783110199680.227.

[87] H. Schwichtenberg. "Constructive analysis with witnesses". Lecture notes. Accessed November 2019. Jan. 2017. URL: http://www.math.lmu.de/~schwicht/seminars/semws16/constr16.pdf.

[88] K. Sojakova. "Higher Inductive Types as Homotopy-Initial Algebras". In: *ArXiv e-prints* (Feb. 2014). arXiv: 1402.0761 [cs.LO].

[89] B. Spitters and E. van der Weegen. "Type classes for mathematics in type theory". In: *Mathematical Structures in Computer Science* 21.4 (2011), pp. 795–825. DOI: 10.1017/S0960129511000119.

[90] P. Taylor. "A lambda calculus for real analysis". In: *J. Logic & Analysis* 2 (2010). Accessed January 2020. URL: http://logicandanalysis.org/index.php/jla/article/view/63/25.

[91] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. Accessed September 2015. Institute for Advanced Study, 2013. URL: https://homotopytypetheory.org/book.

[92] A. S. Troelstra and D. van Dalen. *Constructivism in Mathematics, an Introduction*. Vol. 1. Studies in Logic and the Foundations of Mathematics 121. North-Holland, Aug. 1988. ISBN: 9780444705068.

[93] A. M. Turing. “On Computable Numbers, with an Application to the Entscheidungsproblem”. In: *Proceedings of the London Mathematical Society. Second Series* 42 (1936). See correction [94]., pp. 230–265.

[94] A. M. Turing. “On computable numbers, with an application to the Entscheidungsproblem. A correction”. In: *Proceedings of the London Mathematical Society. Second Series* 43 (1937). See [93]., pp. 544–546.

[95] B. van den Berg and R. Garner. “Types are weak ω-groupoids”. English. In: *Proceedings of the London Mathematical Society* 102.2 (Feb. 2011), pp. 370–394. ISSN: 0024-6115. DOI: 10.1112/plms/pdq026.

[96] A. Vezzosi, A. Mörtberg, and A. Abel. “Cubical agda: a dependently typed programming language with univalence and higher inductive types”. In: *Proc. ACM Program. Lang.* 3.ICFP (2019), 87:1–87:29. DOI: 10.1145/3341691.

[97] S. Vickers. “The localic compact interval is an Escardó-Simpson interval object”. In: *Math. Log. Q.* 63.6 (2017), pp. 614–629. DOI: 10.1002/malq.201500090.

[98] V. Voevodsky, B. Ahrens, D. Grayson, et al. *UniMath — a computer-checked library of univalent mathematics.* Accessed October 2019. URL: https://github.com/UniMath/UniMath.

[99] E. Wiedmer. “Exaktes Rechnen mit reellen Zahlen und anderen unendlichen Objekten”. PhD thesis. ETH Zurich, 1977.