Domain Theory in Constructive and Predicative Univalent Foundations

by

Tom de Jong

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

Submitted: 30 September 2022
Defended: 20 December 2022
Accepted: 1 February 2023

Abstract

We develop domain theory in constructive and predicative univalent foundations (also known as homotopy type theory). That we work predicatively means that we do not assume Voevodsky's propositional resizing axioms. Our work is constructive in the sense that we do not rely on excluded middle or the axiom of (countable) choice. Domain theory studies so-called directed complete posets (dcpos) and Scott continuous maps between them and has applications in a variety of fields, such as programming language semantics, higher-type computability and topology. A common approach to deal with size issues in a predicative foundation is to work with information systems, abstract bases or formal topologies rather than dcpos, and approximable relations rather than Scott continuous functions. In our type-theoretic approach, we instead accept that dcpos may be large and work with type universes to account for this. A priori one might expect that complex constructions of dcpos, involving countably infinite iterations of exponentials for example, result in a need for ever-increasing universes and are predicatively impossible. We show, through a careful tracking of type universe parameters, that such constructions can be carried out in a predicative setting. We illustrate the development with applications in the semantics of programming languages: the soundness and computational adequacy of the Scott model of PCF, and Scott's D_∞ model of the untyped λ-calculus. Both of these applications make use of Escardó's and Knapp's type of partial elements. Taking inspiration from work in category theory by Johnstone and Joyal, we also give a predicative account of continuous and algebraic dcpos, and of the related notions of a small (compact) basis and its rounded ideal completion. This is accompanied by concrete examples, such as the small compact basis of Kuratowski finite subsets of the powerset. The fact that nontrivial dcpos have large carriers is in fact unavoidable and characteristic of our predicative setting, as we explain in a complementary chapter on the constructive and predicative limitations of univalent foundations. We prove no-go theorems for a general class of posets that includes dcpos, bounded complete posets, sup-lattices and frames. In particular, we show that, constructively, locally small nontrivial dcpos necessarily lack decidable equality. Our account of domain theory in univalent foundations has been fully formalised with only a few minor exceptions. The ability of the proof assistant Agda to infer universe levels has been invaluable for our purposes.

Acknowledgements

First and foremost, it is my pleasure to express my deepest thanks to my supervisor Martín Escardó. This thesis simply wouldn't exist without his ideas, patient teaching, and ever-helpful feedback and support. Secondly, I am grateful to my co-supervisor Benedikt Ahrens who really helped me feel at home in Birmingham and has always been available to offer sage advice even after he moved away from Birmingham.

I feel fortunate to have been a part of the very vibrant Theory group at Birmingham and wish to thank all its members, and the participants of our Theory PhD student seminar *Bravo* in particular.

Of the (former) PhD students, Alex Rice, Auke Booij, Ayberk Tosun, Calin Tataru, George Kaye and Todd Waugh Ambridge deserve a special mention. Alex and Calin were excellent teaching assistants and great office mates before they moved to Cambridge. The framework developed to mark *Functional Programming* assignments is largely due to them. Auke was very welcoming and helpful when I first moved to Birmingham. With George I had the pleasure of organising the Theory seminar, upgrading the Theory website and introducing web profiles for PhD students. I have had interesting discussions with Ayberk and Todd and would like to thank them for their efforts as teaching assistants in the modules we taught together. Moreover, like George, they became my friends in Birmingham.

Of the Theory staff members, I would like to mention Anupam Das, Eric Finster, Paul Levy and Sonia Marin in particular. Anupam is responsible for much of the social activities in the Theory group and seemingly never tires of playing devil's advocate. It was a pleasure to work with Eric in *(Advanced) Functional Programming* and at *MGS'22*. Both Sonia and Paul were simply two very friendly presences in the School of Computer Science and I wish to thank them for all our hallway conversations. Moreover, I appreciate that Paul has agreed to chair my viva.

Next, I would like to express my thanks to my thesis group committee members Steve Vickers and Vincent Rahli, and its past members Nicolai Kraus and Noam Zeilberger, for their interest, questions and comments. Furthermore, I wish to thank Vincent for accepting to be my internal examiner.

Last but not least on the list of people at University of Birmingham are Jason Fenemore and his successor Angeliki Bompetsi who must be mentioned for their outstanding work in supporting PhD students and dealing with Worklink.

A special thanks is reserved for Andrea Vezzosi who, upon request, kindly implemented the `lossy-unification` flag in Agda. This heuristic significantly speeds up

Agda's typechecking in some cases and it allowed me to complete my formalisation of domain theory.

I am grateful to Andrej Bauer for agreeing to be my external examiner and for hosting me during my research visit to Ljubljana, where I had many inspiring conversations with Egbert Rijke to whom I am indebted. I quickly felt at home in Ljubljana thanks to them, Ajda Lampe, Alex Simpson, Filip Koprivec, Katja Berčič, Léo Mangel, and in particular, Jure Taslak.

I would also like to thank the organisers, lecturers, participants, and especially, my fellow teaching assistants of the *HoTTEST Summer School 2022*: it has been a great pleasure to learn and work with you.

My fourth and final year as a PhD student was financially supported by the *Homotopy Type Theory Dissertation Fellowship*, funded by Cambridge Quantum Computing and Ilyas Khan. I wish to express my sincere thanks to them and the committee members Steve Awodey, Thierry Coquand, Emily Riehl, and Mike Shulman for allowing me to work on this topic for another year.

I moved to Nottingham towards the end of my PhD and I greatly appreciated Nicolai Kraus' help and support in moving. I am also very grateful to Josh Chen and Stefania Damato for making my first weeks in Nottingham so enjoyable.

I am indebted to Harry Smit for many wonderful conversations, especially when they concerned life as an academic. A special thanks to Menno de Boer for his enthusiasm, our discussions on homotopy type theory and for proofreading parts of this thesis.

These acknowledgements have deliberately been restricted to people that were directly involved in my academic endeavours, and fall short of mentioning many friends and family that I am thankful to. I wish to make a single important exception: Madelon, thank you for all your love, support and understanding. I am deeply grateful that you were, and will be, by my side while I pursue(d) my academic interests abroad.

Contents

Chapter 1

Introduction

Univalent foundations [Voe15], also known as *homotopy type theory* [Uni13] and often abbreviated as *HoTT* or *HoTT/UF*, is a recent enhancement of intensional Martin-Löf Type Theory (MLTT) [Mar84] and has many complementary uses. For example, it is a language for $(\infty, 1)$-toposes [Shu19], it allows for a natural development of synthetic homotopy theory [Uni13; Rij18; BGL+17; BFC+; RBP+] and synthetic group theory [BBC+22; RBP+], and it functions as a modern foundation [Uni13; Esc19b; RBP+] for general mathematics providing an alternative to traditional set-theoretic approaches. Moreover, thanks to the type-theoretic basis of univalent foundations, it is possible to implement proofs in HoTT/UF in proof assistants such as Agda [NDCA+], Cubical Agda [VMA19], Coq [Coq] and Lean [AdMKU+], among others [Jet; ACF+b; ACF+a; ACF+c], allowing for a formalised, computer-checked development of mathematics.

This thesis is concerned with homotopy type theory as a foundations for (formalised) mathematics. Specifically, we develop a formalised account of *domain theory*, an important area in theoretical computer science, in univalent foundations. In fact, we present a fully *constructive* and *predicative* treatment of domain theory within this setting. A precise overview of what is covered can be found in Section 1.2, but for now we emphasise that our development is illustrated and proved to be useful through the exposition of two applications in the semantics of programming languages: the soundness and computational adequacy of the Scott model of PCF [Plo77; Sco93], and Scott's D_∞ model [Sco82b] of the untyped λ-calculus, which are fully formalised in Agda [dJon22a; Har20] and Coq [dJon19; dJon21b].

Domain theory Domain theory [AJ94] studies a particular class of posets and has applications in a variety of fields, such as: programming language semantics [Plo77; Sco82b; Sco93], higher-type computability [LN15] and topology [GHK+03]. For instance, domain-theoretic insights have led to the discovery of surprising algorithms that exhaustively search infinite sets in finite time [Ber90; Esc08]. More generally, domain theory can be used to prove correctness of algorithms through denotational semantics.

The univalent point of view In intensional Martin-Löf Type Theory, the identity type of a type is uniformly and inductively defined. Thus, for every type X, we have a type $x = y$ of identities, or *identifications*. One of the key features of univalent foundations, and of the *univalence axiom* specifically, is that type of identifications captures (in a precise sense) the correct notion of equality, cf. [Uni13, Section 9.8], [CD13], [Esc19b, Section 3.33] and [ANST20]. For example, if we have two elements G and H of the type of groups, then the type $G = H$ is equivalent (in Voevodsky's sense [Voe15]) to the type of group isomorphisms from G to H. In particular, the type of identifications $x = y$ can contain many elements, so equality in univalent foundations is not necessarily a truth value. This naturally leads to higher structures in univalent foundations. In fact, another key insight of Voevodsky [Voe15] was that the stratification of types according to the complexity of their identity types into *(sub)singletons* (truth values), *sets*, 1-groupoids, 2-groupoids, etc. can be internalised and defined inside MLTT.

Consequently, the mathematical distinction between a *property* and (additional) *data* or *structure* is also internalised. Sometimes we know how to express something as an equipment with extra structure, but we are interested in obtaining a property instead. For this, we turn to the *propositional truncation*: the universal method of making a type into a subsingleton. The propositional truncation is an example, and in this thesis the *only* example, of a *higher inductive type* [Uni13, Section 6].

For developing domain theory we typically do not need the theory of higher groupoids. Accordingly, weaker consequences of the univalence axiom (function extensionality and propositional extensionality, to be precise) are often sufficient for our purposes. An important exception, besides its use in the theory of ordinals (Section 6.3.2), is the fundamental notion of $\mathcal{V}$*-smallness*: if we want to know that it expresses a property, then univalence is sufficient and (in some precise sense) necessary, as we explain in Section 2.9. Even if univalence is often not needed, it will hopefully become clear throughout this thesis, that the univalent point of view is prevalent in our work. For example, we recall that if a type X is equipped with a subsingleton-valued reflexive and antisymmetric binary relation, then X is a set, meaning its elements can be equal in at most one way. Moreover, our development fundamentally relies on the aforementioned propositional truncation and also features several applications of a theorem [KECA17, Theorem 5.4] that characterises when we can map from a propositionally truncated type to a set.

Constructivity Constructivity has historically always been important in the type theoretic tradition. Indeed, Martin-Löf invented his type theory to serve as a constructive foundation of mathematics [Mar75]. More recently, its extension, univalent foundations, has been given a computational interpretation through cubical type theory [CCHM18] and this has been implemented in practice as the proof assistant and functional programming language Cubical Agda [VMA19].

That we work constructively means that we do not assume excluded middle, or weaker variants, such as Bishop's LPO [Bis67], or the axiom of choice (which implies excluded middle), or its weaker variants, such as the axiom of countable choice. An advantage of working constructively and not relying on these additional logical axioms is that our development is valid in every $(\infty, 1)$-topos [Shu19] and not just those in which the logic is classical.

Our commitment to constructivity has several manifestations throughout this thesis. For example, it means we cannot simply add a least element to a set to obtain the free pointed *directed complete poset (dcpo)*. Instead of adding a single least element representing an undefined value, we must work with a more complex type of partial elements (Section 3.4). Another example is the distinction between continuity and pseudocontinuity of dcpos (Section 4.5); the notions coincide when the axiom of choice is assumed. Moreover, the absence of countable choice is discussed in connection to semidecidability and the Scott model of PCF in Section 5.2.4.

Constructive mathematics is naturally at home in theoretical computer science, because constructive proofs give rise to algorithms [Bis70; Mar82]. We illustrate this point through a domain-theoretic example. In Section 5.2 we give a constructive proof that the Scott model of PCF is computationally adequate. The constructive nature of the proof yields (in theory, at least) an interpreter: if we can prove that a given program (of base type) is total, then we can compute its numerical outcome through computational adequacy. In classical domain theory it is of fundamental interest [Sco70; Smy77] how its aspects can be formulated in an effective or computational manner. The computational nature of constructive mathematics might enable one to use our constructive development of domain theory to obtain algorithms without having to develop a separate account of effectively given dcpos.

The advent of proof assistants (many based on type theory, including Martin-Löf Type Theory and the Calculus of Constructions [CH88]) has narrowed the gap between mathematics and computer science further and we discuss the implementations of our work in the proof assistants AGDA and COQ below and in more detail in Chapter 7.

Predicativity Our work is predicative in the sense that we do not assume Voevodsky's *resizing* rules [Voe11; Voe15] or axioms. In particular, powersets of small types are large. Before we explain some of the ramifications of this for the domain-theoretic development, we reflect on some of the reasons for working without resizing principles.

First and foremost, it is currently an open problem whether propositional resizing axioms can be given a computational interpretation, as has been done for the univalence axiom and higher inductive types in cubical type theory [CCHM18]. Thus, in line with our constructive agenda and to retain a computational interpretation (in for instance, CUBICAL AGDA [VMA19]) we work in the absence of propositional resizing axioms. Since higher inductive types may be seen as particular resizing principles, it is also noteworthy that the only higher inductive type needed in our work is the propositional truncation. Another reason for being interested in predicativity is the fact that propositional resizing axioms fail in some models of univalent type theory. This is discussed further in Section 1.1.2. Furthermore, it is expected, by analogy to predicative and impredicative set theories, that adding resizing axioms significantly increases the proof-theoretic strength of univalent type theory [Shu19, Remark 1.2]. Lastly, one may have philosophical reservations regarding impredicativity. For example, some constructivists may accept predicative set theories like Aczel's CZF or Myhill's CST, but not Friedman's impredicative set theory IZF [Bee85]. Or, paraphrasing Shulman's narrative [Shu11], one can ask why propositions (or (-1)-types) should be treated differently, i.e. given that we have to take size seriously for n-types for $n > -1$, why not do the same for (-1)-types?

A common approach to deal with domain-theoretic size issues in a predicative foundation is to work with information systems [Sco82a; Sco82b], abstract bases [AJ94] or formal topologies [Sam87; Sam03; CSSV03] rather than dcpos, and approximable relations rather than *Scott continuous functions*. Instead, we work directly with dcpos and Scott continuous functions. In dealing with size issues, we draw inspiration from category theory and make crucial use of type universes and type equivalences to capture *smallness*. For example, in our development of the Scott model of PCF, the dcpos have carriers in the second universe $\mathcal{U}_1$ and least upper bounds for directed families indexed by types in the first universe $\mathcal{U}_0$. Moreover, up to equivalence of types, the order relation of the dcpos takes values in the lowest universe $\mathcal{U}_0$. Seeing a poset as a category in the usual way, we can say that these dcpos are large, but locally small, and have small filtered colimits. The fact that the dcpos have large carriers is in fact unavoidable and characteristic of predicative settings, as proved in Chapter 6. Because the dcpos have large carriers it is a priori not clear that complex constructions of dcpos, involving countably infinite iterations of exponentials for example, do not result in a need for ever-increasing universes and are predicatively possible. We show that they are possible through a careful tracking of type universe parameters, and this is also illustrated by applications, such as the Scott model of PCF and D_∞. Since keeping track of these universes is prone to mistakes, we have implemented much of our work in Agda; its ability to infer universe levels has been invaluable.

Formalisation Type theories are the basis of many successful proof assistants, such as Agda [NDCA+], Coq [Coq] and Lean [AdMKU+]. Much of the work in this thesis has been formalised in Agda using Escardó's TypeTopology [Esc+] development and this has helped considerably to guide our predicative and constructive development of domain theory. A full discussion of the formalisation efforts, including the work in Coq/UniMath, can be found in Chapter 7.

1.1 Related work

We give a brief overview of related work on (constructive and/or predicative) domain theory and of predicativity in general. In short, the distinguishing features of our work are: (i) the adoption of homotopy type theory as a foundation, (ii) a commitment to predicatively and constructively valid reasoning, (iii) the use of type universes to avoid size issues concerning large posets.

1.1.1 Domain theory

The standard works on domain theory, e.g. [AJ94; GHK+03], are based on traditional impredicative set theory with classical logic. A constructive, topos valid, and hence impredicative, treatment of some domain theory can be found in [Tay99, Chapter III].

Domain theory has been studied predicatively in the setting of formal topology [Sam87; Sam03; CSSV03] in [MV04; Neg02; SVV96] and the more recent categorical papers [Kaw17; Kaw21]. In this predicative setting, one avoids size issues by working with information systems [Sco82a; Sco82b], abstract bases [AJ94] or formal topologies, rather than dcpos, and approximable relations rather than Scott continuous functions. Hedberg [Hed96] presented some of these ideas in Martin-Löf Type Theory and formalised them in the proof assistant ALF [Mag94], a precursor to AGDA. A modern formalisation in AGDA based on Hedberg's work was recently carried out in Lidell's master thesis [Lid20].

Our development differs from the above line of work in that it studies dcpos directly and uses type universes to account for the fact that dcpos may be large. An advantage of this approach is that we can work with (Scott continuous) functions rather than the arguably more involved (approximable) relations. For the treatment of continuous (and algebraic) dcpos we turn to the work of Johnstone and Joyal [JJ82] which is situated in category theory where attention must be paid to size issues even in an impredicative setting. In constructive set theory, this corresponds to working with partially ordered classes [Acz06] as opposed to sets, where our notion of a small basis for a dcpo (Section 4.7) is similar to Aczel's notion of a set-generated [Acz06, Section 6.4] dcpo.

Another approach to formalising domain theory in type theory can be found in [BKV09; Doc14]. Both formalisations study ω-chain complete preorders, work with setoids, and make use of COQ's impredicative sort `Prop`. A setoid is a type equipped with an equivalence relation that must be respected by all functions. The particular equivalence relation given by equality is automatically respected of course, but for general equivalence relations this must be proved explicitly. The aforementioned formalisations work with preorders, rather than posets, because they are setoids where two elements x and y are related if $x \leq y$ and $y \leq x$. Our development avoids the use of setoids thanks to the adoption of the univalent point of view. Moreover, we work predicatively and we work with the more general directed families rather than ω-chains, as we intend the theory to be also applicable to topology and algebra [GHK+03].

There are also constructive accounts of domain theory aimed at program extraction [BK09; PM21]. Both these works study ω-chain complete posets (ω-cpos) and define notions of ω-continuity for them. The former [BK09] is notably predicative, but makes use of additional logical axioms: countable choice, dependent choice and Markov's Principle, which are validated by a realisability interpretation. The latter [PM21] uses constructive logic to extract witnesses but employs classical logic in the proofs of correctness by phrasing them in the double negation fragment of constructive logic. By contrast, we study (continuous) dcpos rather than (ω-continuous) ω-cpos and is fully constructive without relying on additional principles such as countable choice or Markov's Principle.

Finally, yet another approach is the field of *synthetic domain theory* [Ros86; Ros87; Hyl91; Reu99; RS99]. Although the work in this area is constructive, it is still impredicative, as it is based on topos logic; but more importantly it has a focus different from that of regular domain theory. The aim is to isolate a few basic axioms and find models in (realisability) toposes where every object is a domain and every morphism is continuous. These models often validate additional axioms, such as Markov's Principle and countable choice, and moreover (necessarily) falsify excluded middle. We have a different goal, namely to develop regular domain theory constructively and predicatively, but in a foundation compatible with excluded middle and choice, while not relying on them or on Markov's Principle or countable choice.

1.1.2 Predicativity

We summarise work on (im)predicativity in univalent foundations as well as work on the limits of predicative mathematics and its relation to the results presented in Chapter 6.

Resizing in models of univalent foundations

As mentioned in the introduction, propositional resizing axioms fail in some models of univalent type theory. A notable example of such a model is Uemura's cubical assembly model [Uem19]. What is particularly striking about Uemura's model is that it does support an impredicative universe $\mathcal{U}$ in the sense that if X is *any* type and $Y : X \to \mathcal{U}$, then $\Pi_{x:X} Y(x)$ is in $\mathcal{U}$ again even if X isn't, but that propositional resizing fails for this universe. We also highlight Swan's (unpublished) results [Swa19b; Swa19a] that show that propositional resizing axioms fail in certain presheaf (cubical) models of type theory. Interestingly, Swan's argument works by showing that the models violate certain collection principles if we assume Brouwerian continuity principles in the metatheory.

By contrast, we should mention that propositional resizing is validated in many models when a classical metatheory is assumed. For example, this is true for any type-theoretic model topos [Shu19, Proposition 11.3]. In particular, Voevodsky's simplical sets model [KL21] validates excluded middle and hence propositional resizing. We note, however, that in other models it is possible for propositional resizing to hold and excluded middle to fail, as shown by [Shu15, Remark 11.24].

Resizing rules versus axioms

This thesis concerns resizing *axioms*, meaning we ask a given type to be equivalent to one in a fixed universe $\mathcal{U}$ of "small" types. Voevodsky [Voe11] originally introduced resizing *rules* which add judgements and hence modify the syntax of the type theory to make the given type inhabit $\mathcal{U}$, rather than only asking for an equivalent copy in $\mathcal{U}$. It is not known whether Voevodsky's resizing rules are consistent with univalent foundations in the sense that no-one has constructed a model of univalent type theory extended with such resizing rules. It is also an open problem [CCHM18, Section 10] whether we have normalisation for cubical type theory extended with resizing rules. In fact, as far as we know, it is an open problem for plain Martin-Löf Type Theory as well.

Limits of predicativity

While Chapters 3 to 5 are devoted to demonstrating the possibility of developing domain theory predicatively in univalent foundations, Chapter 6 instead explores what cannot be done in our predicative setting. Curi had a similar goal and investigated the limits of predicative mathematics in CZF [AR10] in a series of papers [Cur10a; Cur10b; Cur15; Cur18; CR12]. In particular, Curi shows (see [Cur10a, Theorem 4.4 and Corollary 4.11], [Cur10b, Lemma 1.1] and [Cur15, Theorem 2.5]) that CZF cannot prove that various nontrivial posets, including sup-lattices, dcpos and frames, are small. This result is obtained by exploiting that CZF is consistent with the anti-classical generalised uniformity principle (GUP) [vdBer06, Theorem 4.3.5].

Our related Theorem 6.2.21 is of a different nature in two ways. Firstly, the theorem is in the spirit of reverse constructive mathematics [Ish06]: Instead of showing that GUP implies that there are no non-trivial small dcpos, we show that the existence of a non-trivial small dcpo is equivalent to weak propositional resizing, and that the existence of a positive small dcpo is equivalent to full propositional resizing. Thus, if we wish to work with small dcpos, we are forced to assume resizing axioms. Secondly, we work in univalent foundations rather than CZF. This may seem a superficial difference, but a number of arguments in Curi's papers [Cur15; Cur18] crucially rely on set-theoretical notions and principles such as transitive set, set-induction, and the weak regular extension axiom (wREA), which cannot even be formulated in the underlying type theory of univalent foundations. Moreover, although Curi claims that the arguments of [Cur10a; Cur10b] can be adapted to some version of Martin-Löf Type Theory, it is presently not clear whether there is any model of univalent foundations which validates GUP. However, one of the anonymous reviewers of [dJE22a] suggested that Uemura's cubical assemblies model [Uem19] might validate it. In particular, the reviewer hinted that [Uem19, Proposition 21] may be seen as a uniformity principle.

1.2 Outline and summary of contributions

We develop domain theory (Chapter 3) in predicative and constructive univalent foundations (Chapter 2). We include the theory of continuous and algebraic dcpos and rounded ideal completions (Chapter 4), as well as applications in the semantics of programming languages (Chapter 5), namely soundness and computational adequacy of Scott's model of PCF, and Scott's D_∞ model of the untyped λ-calculus. We use type universes to deal with size issues arising in our predicative setting. Moreover, we show that dcpos are predicatively necessarily large in Chapter 6. The development of domain theory, including the applications, is supported by a formalisation, as discussed in Chapter 7. In particular, AGDA's ability to automatically infer universe levels has been invaluable to us.

1.2.1 Summary of contributions

We briefly describe our contributions per chapter and record what parts of this thesis are based on our publications. The full bibliographical details of the publications can be found in Section 1.2.2. Moreover, each chapter features a section at the end with further bibliographical notes.

Chapter 2 *Univalent foundations*

Our exposition of *univalent foundations* is fairly standard and largely follows [Uni13], and in particular [Esc19b]. Two exceptions are Sections 2.11.3 and 2.11.4 which are original contributions where we show small *set quotients* and a *set replacement* principle to be equivalent. Section 2.11, on set quotients, *propositional truncations* and their universe levels, as a whole was included in our work [dJE21b; dJE22a]. Other exceptions are the main results on *indexed* W*-types* with *decidable equality* in Section 2.12 which are due to Jasper Hugunin [Hug17b; Hug17a], and were included in our paper [dJon21b].

Chapter 3 *Basic domain theory*

We present the basic definitions of domain theory: *directed complete posets (dcpos)* and *Scott continuous functions*. It must be remarked that our definitions make use of type universes and are size-aware: we ask for suprema of directed families indexed by types in some fixed universe. We proceed with several basic examples and with constructions of dcpos: *products*, *exponentials*, *lifting* and *bilimits*. Because we work constructively we use Escardó's and Knapp's [EK17; Kna18] lifting monad to construct the free dcpo with a least element on a set. This chapter is a revision of our two papers [dJon21b; dJE21a], see the Notes for further details.

Chapter 4 *Continuous and algebraic dcpos*

This chapter has its roots in [dJE21a], but the treatment has been considerably expanded and revised. In particular, we disentangled the notions of *continuity* and having a *(small) basis* in this thesis. Taking inspiration from the categorical treatment of [JJ82], we give predicatively adequate definitions of continuous and *algebraic* dcpos, and discuss issues related to the absence of the axiom of choice. We also present predicative adaptations of the notions of a basis and the *rounded ideal completion* [AJ94]. Our development is illustrated with several examples: we describe small compact bases for the lifting and the powerset, and consider the ideal completion of the dyadics.

Chapter 5 *Applications in semantics of programming languages*

We describe two applications of domain theory to the *semantics of programming languages*. The first application is a predicative reconstruction of Scott's [Sco72] famous D_∞ model of the untyped λ-calculus, and was included in our paper [dJE21a]. The use of exponentials and bilimits of dcpos is crucial in the construction of D_∞ and we describe how Scott's original proof is adapted to predicative and proof relevant setting of univalent foundations. The second application is the *Scott model* [Plo77; Sco93] of the typed programming language *PCF*, including its *soundness* and *computational adequacy*, and was the subject of our publication [dJon21b]. The Scott model of PCF highlights our use of the lifting monad in particular. We also discuss issues concerning semidecidability and countable choice.

Chapter 6 *Predicativity in order theory*

We complement the above development by exploring the predicative and constructive limits of order theory in univalent foundations. We show that nontrivial dcpos are *necessarily large* and *necessarily lack decidable equality* in our constructive and predicative setting. In particular, the carriers of the dcpos of the Scott model of PCF can only live in the lowest universe $\mathcal{U}_0$ if we work impredicatively. The fact that nontrivial dcpos are necessarily large has the important consequence that *Tarski's theorem* (and similar results) cannot be applied in nontrivial instances, even though it has a predicative proof. Further, we explain, by studying the large sup-lattice of *ordinals*, that generalisations of Tarski's theorem which allow for large structures are provably false. Finally, we elaborate on the connections between requiring suprema of *families* and of *subsets* in our predicative setting. This chapter is taken mostly verbatim from our preprint [dJE22a] which itself is based on our conference paper [dJE21b].

Chapter 7 *Formalisation*

Our development of domain theory in constructive and predicative univalent foundations is accompanied by extensive formalisations that encompass, with very few exceptions, all of Chapters 3 to 5.

1.2.2 Publications

This thesis is based on the following papers, all of which have been published, except for [dJE22a], which has been accepted subject to minor revisions.

[dJE21a] Tom de Jong and Martín Hötzel Escardó. 'Domain Theory in Constructive and Predicative Univalent Foundations'. In: *29th EACSL Annual Conference on Computer Science Logic (CSL 2021)*. Ed. by Christel Baier and Jean Goubault-Larrecq. Vol. 183. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2021, 28:1–28:18. DOI: `10.4230/LIPIcs.CSL.2021.28`. Expanded version with full proofs available on arXiv: `2008.01422 [math.LO]`.

[dJE21b] Tom de Jong and Martín Hötzel Escardó. 'Predicative Aspects of Order Theory in Univalent Foundations'. In: *6th International Conference on Formal Structures for Computation and Deduction (FSCD 2021)*. Ed. by Naoki Kobayashi. Vol. 195. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2021, 8:1–8:18. DOI: `10.4230/LIPIcs.FSCD.2021.8`.

[dJE22a] Tom de Jong and Martín Hötzel Escardó. 'On Small Types in Univalent Foundations'. Sept. 2022. arXiv: `2111.00482 [cs.LO]`. Revised and expanded version of [dJE21b]. Accepted pending minor revision to a special issue of *Logical Methods in Computer Science* on selected papers from *FSCD 2021*.

[dJon21b] Tom de Jong. 'The Scott model of PCF in univalent type theory'. In: *Mathematical Structures in Computer Science* 31.10 (2021): *Homotopy Type Theory 2019*, pp. 1270–1300. DOI: `10.1017/S0960129521000153`.

The work [dJE21b] won the *Best Paper by a Junior Researcher* award and some of the above publications led to the award of the *Homotopy Type Theory Dissertation Fellowship*.

Chapter 2

Univalent foundations

Our foundational starting point is intensional Martin-Löf Type Theory (MLTT) [Mar75; Mar84] with an empty type $\mathbf{0}$, unit type $\mathbf{1}$, natural numbers type $\mathbb{N}$, binary coproducts $(+)$, dependent sums (Σ), dependent products (Π), intensional identity types, and general inductive types (i.e. W-types as discussed in Section 2.12).

In the upcoming sections, we will discuss additions to this type theory that will define our foundational setup. These additions will be: universes, function extensionality, propositional extensionality, propositional truncations, and (sometimes) univalence.

We also introduce some of the characteristic features of univalent foundations, such as the stratification of types into (sub)singletons, sets, 1-groupoids, …, according to the complexity of their identity types (Section 2.3), the notions of embedding and equivalence (Section 2.4), propositional truncations (Section 2.6) and univalence (Section 2.8).

The notion of subsingletons gives rise to a refinement of the Curry–Howard paradigm to logical-propositions-as-subsingletons, as explained in Section 2.7. This distinguishes univalent foundations from other type theories such as the *Calculus of (Inductive) Constructions* [CH88] and Coq [Coq], where logic is set to take place in a special designated `Prop` type. The fact that our logic is constructive is discussed in Section 2.7.3, while its predicativity is studied in Section 2.10 after we introduce the theory of (locally) small types in Section 2.9. Finally, set quotients and set replacement are examined in a predicative context in Section 2.11.

Notation If X is a type and $Y(x)$ a dependent type over X, then we denote its type of dependent functions as $\Pi_{x:X}Y(x)$ or sometimes just ΠY, and similarly for Σ-types. If $Y(x)$ does not depend on X, then we respectively denote $\Pi_{x:X}Y(x)$ by $X \to Y$ and $\Sigma_{x:X}Y(x)$ by $X \times Y$.

Further, the first projection is denoted by $\mathrm{pr}_1 : (\Sigma_{x:X}Y(x)) \to X$, while the second projection is called $\mathrm{pr}_2 : \Pi_{s:\Sigma_{x:X}Y(x)}Y(\mathrm{pr}_1(s))$. The identity map on a type X is denoted by id or id_X and function composition is denoted by $g \circ f$ where the codomain of f is definitionally equal to the domain of g.

Definitional (or judgemental) equality is denoted by $\equiv$ and we use $:\equiv$ to signal that we are making a definition. For a type X with elements $x, y : X$, the corresponding identity type is denoted by $x = y$, or sometimes $x =_X y$ to highlight the type of x and y. If $p : x = y$, then we write $p^{-1} : y = x$ for its inverse (up to intensional equality) and if $q : y = z$, then we write $p \bullet q : x = z$ for the composition of the identifications. Moreover, we sometimes find it convenient to write $f \sim g$ for $\Pi_{x:X} f(x) = g(x)$ for (dependent) functions $f, g : \Pi_{x:X} Y(x)$.

The unique element of the unit type will be written as $\star : \mathbf{1}$. The constructors of the coproduct are denoted by inl and inr. We denote the particular coproduct $\mathbf{1} + \mathbf{1}$ by $\mathbf{2}$ and write 0 and 1 for its elements.

Terminology Following [Uni13], when we define a (dependent) function using the elimination rule of the identity type, then we say we define it by *(path) induction*. We will also refer to elements of the identity type as "equalities", "identifications" or "paths", and write refl or refl_x for reflexivity at x, the canonical element of $x = x$.

In line with the Curry–Howard paradigm and [Uni13], we will work informally in type theory and for example say that "$Y(x)$ holds for every $x : X$" if we have an element of $\Pi_{x:X} Y(x)$. Similarly, "P and Q hold" will mean that the type $P \times Q$ has an element.

2.1 Type universes

Type universes will play a fundamental role in our development of domain theory in a predicative context, because, by our definitions, they will keep track of size for us. But type universes are indispensable in MLTT anyway as they have many other uses, e.g. defining type families by induction and collecting mathematical structures into a single type (e.g. the type of all (small) groups). Our setup of type universes follows that of Agda [NDCA+] and [Esc19b; Esc21], but differs from that of Coq [Coq] and [Uni13] because we do not assume cumulativity (see Remark 2.1.1); any lifting of types to higher universes will be annotated explicitly (see Remark 2.1.2).

Intuitively a type universe is a type of types. In so-called Tarski-style universes, the elements of a universe $\mathcal{U}$ are codes for types and the universe comes equipped with a decoding τ such that if $x : \mathcal{U}$, then $\tau(x)$ is a type. This is useful, because it maintains a clean separation between types and elements of types, but cumbersome in practice. By contrast, in Russell-style universes, the elements of a universe $\mathcal{U}$ are actual types. This complicates the meta-theory because now $X : \mathcal{U}$ is both an element of $\mathcal{U}$ and a type. Our universes will be presented à la Russell, but one should read this as an abbreviation for Tarski-style universes.

2.1.1 Operations on universes

First of all, we postulate that there is a universe $\mathcal{U}_0$. Secondly, we postulate two meta-operations on universes: a unary operation $(-)^+$, called *successor*, and a binary operation $(-) \sqcup (-)$ satisfying the following conditions:

(i) for every universe $\mathcal{U}$, we have $\mathcal{U}_0 \sqcup \mathcal{U} \equiv \mathcal{U}$ and $\mathcal{U} \sqcup \mathcal{U}^+ \equiv \mathcal{U}^+$;
(ii) the operation $(-) \sqcup (-)$ is definitionally idempotent, commutative and associative, i.e. for all universes $\mathcal{U}$, $\mathcal{V}$ and $\mathcal{W}$, we assume $\mathcal{U} \sqcup \mathcal{U} \equiv \mathcal{U}$ as well as $\mathcal{U} \sqcup \mathcal{V} \equiv \mathcal{V} \sqcup \mathcal{U}$ and $(\mathcal{U} \sqcup \mathcal{V}) \sqcup \mathcal{W} \equiv \mathcal{U} \sqcup (\mathcal{V} \sqcup \mathcal{W})$;
(iii) the successor operation $(-)^+$ distributes over $(-) \sqcup (-)$ definitionally, i.e. for every two universes $\mathcal{U}$ and $\mathcal{V}$, we have $(\mathcal{U} \sqcup \mathcal{V})^+ \equiv \mathcal{U}^+ \sqcup \mathcal{V}^+$.

In particular, we can iterate the successor operation starting with $\mathcal{U}_0$ to obtain an infinite tower of universes that we denote by $\mathcal{U}_0, \mathcal{U}_1, \mathcal{U}_2, \ldots$.

Remark 2.1.1. We do *not* assume cumulativity of the universes, i.e. we do not require that $A : \mathcal{U}$ implies $A : \mathcal{U} \sqcup \mathcal{V}$ for every two universes $\mathcal{U}$ and $\mathcal{V}$. However, in Remark 2.1.2 we describe how we can easily transport types to higher universes in a suitable sense.

2.1.2 Closure properties

We assume the following closure properties regarding universes:
(i) if $X : \mathcal{U}$, then the identity type $(x = y)$ lives in $\mathcal{U}$ for every $x, y : X$;
(ii) if $X : \mathcal{U}$ and $Y : \mathcal{V}$, then $X + Y : \mathcal{U} \sqcup \mathcal{V}$;
(iii) if $X : \mathcal{U}$ and $Y : X \to \mathcal{V}$, then the types $\Sigma_{x:X} Y(x)$ and $\Pi_{x:X} Y(x)$ are both assumed to be in $\mathcal{U} \sqcup \mathcal{V}$;
(iv) the universe $\mathcal{U}_0$ contains the type of natural numbers $\mathbb{N}$;
(v) every universe $\mathcal{U}$ contains copies $\mathbf{0}_{\mathcal{U}}$ and $\mathbf{1}_{\mathcal{U}}$ of respectively the empty and unit type.

We write $\mathbf{2}_{\mathcal{U}} :\equiv \mathbf{1}_{\mathcal{U}} + \mathbf{1}_{\mathcal{U}}$ and 0 and 1 for its two inhabitants.

Remark 2.1.2. To compensate for the fact that we do not assume cumulativity, we observe that, using the empty and unit types, it is easy to define a map

$$\mathrm{lift}_{\mathcal{U},\mathcal{V}} : \mathcal{U} \to \mathcal{U} \sqcup \mathcal{V}$$

for every two universes such that every type $X : \mathcal{U}$ is equivalent (a notion we define later) to $\mathrm{lift}_{\mathcal{U},\mathcal{V}}(X)$. For instance, the map $X \mapsto X + \mathbf{0}_{\mathcal{V}}$ does the job, as does $X \mapsto X \times \mathbf{1}_{\mathcal{V}}$. But, in the absence of cumulativity, the types X and $\mathrm{lift}_{\mathcal{U},\mathcal{V}}(X)$ cannot be *equal*, because they do not even live in the same universe.

2.2 Identity types and function extensionality

The identity type is defined uniformly for every type $X : \mathcal{U}$ as the inductive family $X \to X \to \mathcal{U}$ generated by $\mathrm{refl} : x = x$. It is possible, however, to show that the identity type acts as expected for specific types. For example, given $(x, y), (x', y') : X \times Y$, we would expect $(x, y) = (x', y')$ to hold precisely when $x = x'$ and $y = y'$. Similarly, we can show that if $x : X$ and $y : Y$, then $\mathrm{inl}(x) =_{X+Y} \mathrm{inr}(y)$ never holds, while $\mathrm{inl}(x) =_{X+Y} \mathrm{inl}(x')$ holds precisely when $x =_X x'$. The situation for Σ-types is slightly more involved and requires the notion of *transport*.

Definition 2.2.1 (Transport)**.** For every type $X : \mathcal{U}$ and type family $Y : X \to \mathcal{V}$, we have a function $\mathrm{transport}^Y : \Pi_{x,x':X}(x = x' \to Y(x) \to Y(x'))$ defined inductively as $\mathrm{transport}^Y(\mathrm{refl}) :\equiv \mathrm{id}$, where we have left the arguments x and x' implicit.

We also take this opportunity to define the action of a map on paths.

Definition 2.2.2 (Action on paths, ap_f)**.** Every function $f : X \to Y$ induces a map on identity types $\mathrm{ap}_f : (x = y) \to (f(x) = f(y))$ for every $x, y : X$ defined inductively by $\mathrm{ap}_f(\mathrm{refl}) :\equiv \mathrm{refl}$, and sometimes called the *action (of f) on paths.*

For characterising identity types, we introduce the notion of an invertible map. In Section 2.4 we consider the more refined notion of a map being an *equivalence.* Another useful notion is that of a left-cancellable map, which we will similarly refine to the notion of an *embedding* later.

Definition 2.2.3 (Invertibility and left-cancellability)**.** A map $f : X \to Y$ is

(i) *invertible* if we have a specified $g : Y \to X$ with $g(f(x)) = x$ for every $x : X$ and $f(g(y)) = y$ for every $y : Y$, and

(ii) *left-cancellable* if for every $x, x' : X$, we have a function

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

Lemma 2.2.4. *Every invertible map is left-cancellable.*

Proof. If $f : X \to Y$ is invertible with inverse $g : Y \to X$, then for every $x, x' : X$, we have $f(x) = f(x') \xrightarrow{\mathrm{ap}_g} g(f(x)) = g(f(x')) \to x = x'$, where the final map is obtained using that g is the inverse of f. □

Lemma 2.2.5. *For every type family Y over a type X, and every $(x, y), (x', y') : \Sigma Y$, we have invertible maps between the identity type $(x, y) =_{\Sigma Y} (x', y')$ and the Σ-type $\Sigma_{p:x=x'} \mathrm{transport}^Y(p, y) = y'$.*
In particular, if Y is just a type, then we have invertible maps between the identity type $(x, y) =_{X \times Y} (x', y')$ and the product of identity types $(x = x') \times (y = y')$.

The need for transporting y arises from the fact that $y : Y(x)$, while $y' : Y(x')$, so y and y' cannot be equal as they do not have the same type.

Proof. The invertible maps are inductively defined as

$$\begin{array}{ccc} (x, y) =_{\Sigma Y} (x', y') & & \Sigma_{p:x=x'} \mathrm{transport}^Y(p, y) = y' \\ \mathrm{refl}_{(x,y)} & \mapsto & (\mathrm{refl}_x, \mathrm{refl}_y) \\ \mathrm{refl}_{(x,y)} & \leftarrow\!\shortmid & (\mathrm{refl}_x, \mathrm{refl}_y) \end{array}$$

□

By contrast, it is not provable in intensional Martin-Löf Type Theory that two pointwise equal functions are equal, as shown by [Str93, Theorem 3.17]. Therefore, we wish to add it as an axiom. However, for reasons that we will explain later, the official formulation of the axiom will have to wait. Even so, the official formulation will be logically equivalent to the following unofficial axiom that we introduce now under the name "naive function extensionality".

Definition 2.2.6 (Naive function extensionality)**.** *Naive function extensionality* asserts that for every two functions $f, g : X \to Y$, if $f(x) = g(x)$ for every $x : X$, then $f = g$. In other words, naive function extensionality says that pointwise functions are equal.

The reason that we introduce naive function extensionality this early is that it allows us to present many useful results earlier. When using it, we typically drop the word "naive" and simply say "by function extensionality". The following lemma prepares us for the official formulation of function extensionality later.

Lemma 2.2.7. *Naive function extensionality is logically equivalent to all of the below, seemingly stronger, statements:*

(i) for every two dependent *functions $f, g : \Pi_{x:X} Y(x)$, if $f(x) = g(x)$ for every $x : X$, then $f = g$;*
(ii) for every two functions $f, g : X \to Y$, the canonical function from $f = g$ to $\Pi_{x:X} f(x) = g(x)$ given by $e \mapsto \lambda x \,.\, \mathrm{ap}_{\lambda h.h(x)}(e)$ is invertible;
(iii) for every two dependent *functions $f, g : \Pi_{x:X} Y(x)$, the canonical function from $f = g$ to $\Pi_{x:X} f(x) = g(x)$ is invertible.*

Proof. See [Esc19b, Section 3.18]. □

2.3 Subsingletons, sets and (higher) groupoids

A fundamental idea in univalent foundations is the stratification of types according to the complexity of their identity types.

Definition 2.3.1 (Subsingleton, proposition, truth value)**.** A type X is a *subsingleton* (or *proposition* or *truth value*) if it has at most one element, meaning we have an element of $\Pi_{x,y:X}\, x = y$.

Remark 2.3.2 (Property and data). If a type X is a subsingleton, then we like to say that X is *property*. By contrast, if X can have more than one element, then we sometimes say that X is *data*. For example, as we explain in Section 2.4 the notion of being an equivalence is a property, while being invertible is data. Another example comes from our development of domain theory and is the distinction between structural continuity and continuity of a directed complete poset (Section 4.4). The former equips the poset with additional structure in the form of a specified mapping, while the latter only requires some unspecified mapping to exist, in a sense to be made precise in Section 2.6.

The names "proposition" and "truth value" suggests that subsingletons are related to logic and indeed we will use the subsingletons to encode logic in our type theory in Section 2.7.

Definition 2.3.3 (Type of subsingletons, $\Omega_{\mathcal{U}}$)**.** The *type of subsingletons* in a universe $\mathcal{U}$ is defined as $\Omega_{\mathcal{U}} :\equiv \Sigma_{P:\mathcal{U}}$ is-subsingleton(P).

Definition 2.3.4 (Singleton, contractibility)**.** A type X is a *singleton* (or said to be *contractible*) if it is a subsingleton and moreover we have an element of X.

Theorem 2.3.5. *For every element x of a type X, the type $\Sigma_{y:X}\, x = y$ is a singleton with unique element* (x, refl).

Proof. We have to show that for every $y : X$ and $p : x = y$, the pair $(y, p) = (x, \mathrm{refl})$, but by path induction we may assume that $y \equiv x$ and $p \equiv \mathrm{refl}$ in which case it is trivial. □

Example 2.3.6. The empty type $\mathbf{0}_{\mathcal{U}}$ and the unit type $\mathbf{1}_{\mathcal{U}}$ in any universe $\mathcal{U}$ are both subsingletons. A further example of a subsingleton is the type

$$\Sigma_{n:\mathbb{N}}(n \text{ is the least number } k \text{ for which } \alpha_k = 0)$$

where $\alpha : \mathbb{N} \to \mathbf{2}$. By contrast, the type $\Sigma_{n:\mathbb{N}}\, \alpha_n = 0$ is not necessarily a subsingleton, because α could have multiple roots.

Remark 2.3.7. The fact that the type $\Sigma_{n:\mathbb{N}}(n \text{ is the least number } k \text{ for which } \alpha_k = 0)$ from Example 2.3.6 is a subsingleton shows us that subsingletons are not necessarily proof irrelevant, because an inhabitant of that type gives us an explicit natural number. Another example that we will discuss in some detail later (Section 2.4) is the notion of an equivalence: the type expressing that a map f is an equivalence is a subsingleton, but given an inhabitant of it, we can construct an inverse of f.

So a subsingleton is a type where any two elements are equal. Going up one level, we consider types where any two identifications are equal.

Definition 2.3.8 (Set)**.** A type X is a *set* if the type $x = y$ is a subsingleton for every $x, y : X$.

In other words, in a set two elements are equal in at most one way.

Example 2.3.9. Every type with decidable equality is a set. This classic result is known as Hedberg's Theorem [Hed98] (Theorem 2.7.12 below). In particular, the type $\mathbb{N}$ of natural numbers is a set.

We could iterate these definitions and arrive at higher *groupoids*: a 1-groupoid is a type whose identity types are sets, a 2-groupoid is a type whose identity types are 1-groupoids, etc.

Example 2.3.10. An example of a type that is not a set is the circle $\mathbb{S}^1$, a higher inductive type [Uni13, Chapter 6] with a chosen basepoint base : $\mathbb{S}^1$ for which we can prove, assuming univalence (see Definition 2.8.1), that (base = base) is equivalent to $\mathbb{Z}$, the type of the integers. In our work, we will not assume any higher inductive types other than propositional truncations.

As explained in [Uni13, Example 3.1.9], another example of a non-set is given by any univalent universe $\mathcal{U}$. By univalence of $\mathcal{U}$, one can show that $\mathbf{2}_{\mathcal{U}} = \mathbf{2}_{\mathcal{U}}$ contains exactly two elements, which shows that the universe $\mathcal{U}$ is not a set. A nice example of a 1-groupoid is the type of groups in a universe $\mathcal{U}$: for two groups G and H, the type $G =_{\mathrm{Grp}_{\mathcal{U}}} H$ is, assuming univalence, equivalent to the type of group isomorphisms between G and H, which is a set.

In this thesis we do not need to develop the theory of higher groupoids and besides universes we can often restrict our attention to sets and subsingletons, like in the upcoming closure results.

2.3.1 Hedberg's Lemma

In proving various results about subsingletons and sets, the following lemma, which we call Hedberg's Lemma, proves highly useful. While all the techniques were already present in Hedberg's paper [Hed98], the precise formulation presented below only appeared in [KECA17, Lemma 3.11].

Definition 2.3.11 (Constant)**.** A map $f : X \to Y$ is *constant* if $f(x) = f(x')$ for every $x, x' : X$.

Remark 2.3.12. This is sometimes called *weakly constant* or *wildly constant*, because if Y is not a set, then f can be constant in more than one way, but also in an incoherent way in a precise higher categorical sense [Kra15]. In other words, the above definition does not account for further coherence conditions. But we will only be interested in constant maps to sets, so we simply stick to "constant".

Lemma 2.3.13 (Hedberg's Lemma)**.** *Let x be an arbitrary, but fixed element of a type X. If we have a constant endofunction on $x = y$ for every $y : X$, then $x = y$ is a proposition for every $y : X$.*

Proof. Suppose that $f_y : (x = y) \to (x = y)$ is constant for every $y : X$. By induction on $p : x = y$, we see that every $p : x = y$ is equal to $f_x(\mathrm{refl})^{-1} \bullet f_y(p)$. Hence, if $y : X$ is arbitrary and $p, q : x = y$, then $p = f_x(\mathrm{refl})^{-1} \bullet f_y(p) = f_x(\mathrm{refl})^{-1} \bullet f_y(q) = q$, as f_y is constant. Hence, each $x = y$ is a proposition, as desired. ☐

2.3.2 Closure properties

The proofs of the following two lemmas illustrate how to apply Hedberg's Lemma.

Lemma 2.3.14. *Every subsingleton is a set.*

Proof. If X is a subsingleton, then for every $x, y : X$ we have a map $\mathbf{1} \to (x = y)$. But the composite $(x = y) \to \mathbf{1} \to (x = y)$ is constant, because $\mathbf{1}$ is a subsingleton, so X must be a set by Hedberg's Lemma. □

Lemma 2.3.15. *If Y is a subsingleton (or set, respectively) and $f : X \to Y$ is a left-cancellable map, then X is a subsingleton (or set, respectively) too. In particular, this holds if f is invertible.*

Proof. Assume that f is left-cancellable. Suppose first that Y is a subsingleton and let $x, x' : X$ be arbitrary. Then $f(x) = f(x')$ because Y is a subsingleton, but f is left-cancellable so we get the desired $x = x'$, showing that X is a subsingleton.
Now suppose that Y is a set. To show that X is a set, it suffices, by Hedberg's Lemma, to construct a constant endofunction on $x = x'$ for every $x, x' : X$. But because Y is a set, the second map, and hence the composite

$$x = x' \xrightarrow{\mathrm{ap}_f} f(x) = f(x') \xrightarrow{f \text{ is left-cancellable}} x = x'$$

is constant.
The final claim holds because every invertible map is left-cancellable as shown in Lemma 2.2.4. □

Theorem 2.3.16. *The subsingletons and sets are closed under Σ, e.g. if X is a subsingleton and Y is a type family over X such that each $Y(x)$ is a subsingleton, then $\Sigma_{x:X} Y(x)$ is a subsingleton too. In particular, if Y is just a type, then $X \times Y$ is a subsingleton (or set, respectively) if both X and Y are.*

Proof. Suppose first that X and each $Y(x)$ are subsingletons and that we have two pairs $(x, y), (x', y') : \Sigma Y$. We wish to show that $(x, y) = (x', y')$. By Lemma 2.2.5 it suffices to find an element $p : x = x'$ and an element of $\mathrm{transport}^Y(p, y) = y'$. But X is assumed to be a subsingleton, so we have such a p and moreover, $Y(x')$ is assumed to be a proposition, so any two of its element are equal, in particular $\mathrm{transport}^Y(p, y)$ and y' must be equal.
Now suppose that X and each $Y(x)$ are sets. To show that ΣY is a set, we have to prove that $(x, y) = (x', y')$ is a subsingleton for every two pairs $(x, y), (x', y') : \Sigma Y$. By Lemmas 2.2.5 and 2.3.15 it is enough to show that $\Sigma_{p:x=x'} \mathrm{transport}^Y(p, y) = y'$ is a subsingleton. But this is a Σ-type of subsingletons because X and $Y(x')$ are assumed to be sets and we already proved that such Σ-types are subsingletons again. □

The proof of the following fundamental theorem features our first application of function extensionality. That function extensionality is in fact necessary is discussed in [Esc19b, Section 3.18].

Theorem 2.3.17. *The subsingletons and sets form a (dependent) exponential ideal. That is, if Y is a type family over an* arbitrary *type X such that each $Y(x)$ is a subsingleton (or set, respectively), then $\Pi_{x:X} Y(x)$ is a subsingleton (or set, respectively) too.*

In particular, if Y is just a type, then $X \to Y$ is a subsingleton (or set, resp.) if Y is.

We stress that, unlike for Σ-types, X is *not* required to be a subsingleton or a set.

Proof. Note that if each $Y(x)$ is a subsingleton, then $f(x) = g(x)$ for all $x : X$ and all functions $f, g : \Pi Y$. Hence, $f = g$ for all $f, g : \Pi_{x:X} Y(x)$ by function extensionality. Now assume that each $Y(x)$ is a set and let $f, g : \Pi_{x:X} Y(x)$ be arbitrary. We must show that $f = g$ is a subsingleton. By function extensionality and Lemma 2.2.7 we have an invertible map from $f = g$ to $\Pi_{x:X} f(x) = g(x)$ for every two $f, g : \Pi_{x:X} Y(x)$. Hence, by Lemma 2.3.15 it suffices to prove that $\Pi_{x:X} f(x) = g(x)$ is a subsingleton. But each $Y(x)$ is a set, so this is a Π-type over a subsingleton-valued family and hence a subsingleton itself as we have just shown. □

Lemma 2.3.18. *For every type X, if we have a function $X \to \text{is-subsingleton}(X)$, then X is a subsingleton.*

Proof. Suppose that we have a function $f : X \to \text{is-subsingleton}(X)$. To show that X is a subsingleton, recall that $\text{is-subsingleton}(X) :\equiv \Pi_{x,y:X}(x = y)$. So let $x : X$ be arbitrary and note that we must prove that $x = y$ for every $y : X$. But this is given by $f_x(x)$. □

Theorem 2.3.19. *Being a set or (sub)singleton is a property, i.e. for every type X, the types $\text{is-subsingleton}(X)$, $\text{is-singleton}(X)$ and $\text{is-set}(X)$ are themselves subsingletons.*

Proof. We first show that $\text{is-subsingleton}(X)$ is a subsingleton. By Lemma 2.3.18 we may assume that we have an element of $\text{is-subsingleton}(X)$, i.e. that X is a subsingleton. Now recall that $\text{is-subsingleton}(X) :\equiv \Pi_{x,y:X}(x = y)$. By Theorem 2.3.17 it suffices to prove that $x = y$ is a subsingleton for every $x, y : X$. But this is indeed the case, as X was assumed to be a subsingleton, and hence must be a set by Lemma 2.3.14. Thus, for every type X, the type $\text{is-subsingleton}(X)$ is a subsingleton, as desired. In particular, if X is an arbitrary type, then $\text{is-subsingleton}(x = y)$ is a subsingleton for every $x, y : X$. Hence, $\text{is-set}(X)$ is a subsingleton by Theorem 2.3.17. Finally, to show that $\text{is-singleton}(X)$ is a subsingleton, note that we can assume that X is a singleton by Lemma 2.3.18. Hence, in particular, it is a subsingleton, so $\text{is-singleton}(X) \equiv X \times \text{is-subsingleton}(X)$ is seen to be a proposition by Theorem 2.3.16. □

2.3.3 Propositional extensionality

Having introduced subsingletons (or propositions), we ask: when should two propositions be equal? Since they have at most one element, it seems natural to want them to be equal exactly when one has an element if and only if the other does.

Definition 2.3.20 (**Axiom**: Propositional extensionality)**.** *Propositional extensionality* asserts that for every two propositions P and Q, if $P \to Q$ and $Q \to P$, then $P = Q$. In other words, it says that logically equivalent propositions are equal.

We take propositional extensionality as an axiom and often use it tacitly.

Remark 2.3.21. This really is an axiom *scheme*: we add propositional extensionality for propositions in every type universe $\mathcal{U}$. One is forced to formulate the axiom for universes anyway, because writing $P = Q$ only makes sense when P and Q are elements of the same type, which we are taking to be a universe $\mathcal{U}$ here.

Theorem 2.3.22. *Assuming function extensionality and propositional extensionality for $\mathcal{U}$, the type $\Omega_{\mathcal{U}}$ of propositions in $\mathcal{U}$ is a set.*

Proof. The type $\Omega_{\mathcal{U}}$ consists of pairs (P, i) with $P : \mathcal{U}$ a type and i : is-subsingleton(P). By Theorem 2.3.19, two such pairs are equal if and only if their first components are equal, so this is what we set out to prove.
By Hedberg's Lemma (Lemma 2.3.13), it suffices to construct a constant endomap on $P = Q$ for every two propositions P and Q in $\mathcal{U}$. But the map

$$(P = Q) \xrightarrow{\mathrm{refl} \mapsto (\mathrm{id},\mathrm{id})} (P \to Q) \times (Q \to P) \xrightarrow{\text{prop-ext}_{\mathcal{U}}} (P = Q)$$

does the job, because the type $(P \to Q) \times (Q \to P)$ is a subsingleton by Theorems 2.3.16 and 2.3.17. □

Proposition 2.3.23 (Propositional extensionality is a property). *Assuming only function extensionality, the type expressing propositional extensionality for a universe $\mathcal{U}$ is a subsingleton.*

Proof. Lemma 2.3.18 tells us that we may assume propositional extensionality for $\mathcal{U}$ to prove the lemma. By a repeated application of Theorem 2.3.17, it suffices to prove that $P = Q$ is a proposition for every two propositions P and Q in $\mathcal{U}$. But this is given by Theorem 2.3.22. □

Remark 2.3.24. It is important that propositional extensionality is a property, because it gives us a guarantee that a construction using propositional extensionality cannot depend on a specific witness of propositional extensionality as they are all the same up to intensional equality. Put differently, in adding it as an axiom to our type theory we are adding a property and not additional data.

2.4 Embeddings, equivalences and retracts

A major application of having defined the notion of (sub)singleton is being able to define what it means for a map to be an embedding or an equivalence.

Definition 2.4.1 (Fibre, fib_f). The *fibre* of a map $f : X \to Y$ at $y : Y$ is the type $\mathrm{fib}_f(y) :\equiv \Sigma_{x:X}\, f(x) = y$.

Definition 2.4.2 (Embedding, equivalence, $X \hookrightarrow Y$, $X \simeq Y$)**.** A map $f : X \to Y$ is

(i) an *embedding* if all of its fibres are subsingletons, and

(ii) an *equivalence* if all of its fibres are singletons.

We denote the type of embeddings from X to Y by $X \hookrightarrow Y$ and the type of equivalences from X to Y by $X \simeq Y$.

We understand this definition as follows: a map $f : X \to Y$ is an equivalence if for every $y : Y$ there is exactly one $x : X$ with $f(x) = y$. Similarly, a map $f : X \to Y$ is an embedding if for every $y : Y$ there is at most one $x : X$ with $f(x) = y$.

Observe how the notions of embedding and equivalence and defined in terms of the fibres of the map. After we introduce the proposition truncation in Section 2.6, we will see that a map is a surjection if its fibres are all inhabited and a split surjection if its fibres are all pointed. Thus, the fibres of a map are of fundamental interest.

Theorem 2.4.3. *Being an embedding/equivalence is a property, i.e. for every map $f : X \to Y$ the types expressing that f is an embedding/equivalence are subsingletons.*

Proof. Immediate consequence of Theorems 2.3.17 and 2.3.19. □

It is natural to wonder if, in the definition of a map being an equivalence, the uniqueness conditions can be expressed as: "if we have another $x' : X$ such that $f(x') = y$, then $x' = x$". This is equivalent to the above definition when Y is a set, but in general this fails to account for the structure that the identity types of Y might carry in the sense that the types expressing this may fail to be propositions. The upshot of Theorem 2.4.3 is that the equivalence between X and Y form a subtype of all functions from X to Y, as we explain in Example 2.4.10 below.

The following result is rather useful for proving that a map is an equivalence.

Proposition 2.4.4. *A map $f : X \to Y$ is an equivalence if and only if it is invertible.*

Proof. See [Esc19b, Section 3.10] for a proof or [Uni13, Chapter 4] where this and related results and issues are discussed at length. □

One may ask why we did not define f to be an equivalence when it's invertible. After all, Proposition 2.4.4 tells us that the two are logically equivalent. However, they are *not* equivalent in the sense of Definition 2.4.2, as being invertible may fail to be property (recall Remark 2.3.2), cf. [Uni13, Theorem 4.1.3], while being an equivalence is. Without going into the details of the proof too much, the circle $\mathbb{S}^1$ (recall Example 2.3.10) provides an illustrative example where invertibility may fail to be a property, as the following example makes clear.

Example 2.4.5 (Invertibility is not necessarily a property)**.** The type expressing that the identity on $\mathbb{S}^1$ is invertible is

$$\Sigma_{f:\mathbb{S}^1\to\mathbb{S}^1}(\Pi_{x:\mathbb{S}^1}(f \circ \mathrm{id})(x) = \mathrm{id}(x)) \times (\Pi_{x:\mathbb{S}^1}(\mathrm{id} \circ f)(x) = \mathrm{id}(x)).$$

But by function extensionality, see Lemma 2.2.7, this is equivalent to

$$\Sigma_{f:\mathbb{S}^1\to\mathbb{S}^1}(f = \mathrm{id}) \times (f = \mathrm{id}),$$

which, by Theorem 2.3.5, is equivalent to $\mathrm{id}_{\mathbb{S}^1} = \mathrm{id}_{\mathbb{S}^1}$. By function extensionality, this is equivalent to $\Pi_{x:\mathbb{S}^1}\, x = x$ which can be shown to be equivalent to $(\mathrm{base} = \mathrm{base})$ and hence to the type of integers $\mathbb{Z}$, see Example 2.3.10.

As mentioned above, the fact that being an equivalence is a property ensures that the type of equivalences is a subtype (in the sense of Definition 2.4.7 below). Moreover, the notion of an equivalence is crucial for formulating the univalence axiom (Definition 2.8.1 below), because the formulation using invertible maps is provably false [Uni13, Exercise 4.6(c)].

Proposition 2.4.6. *A map $f : X \to Y$ is an embedding if and only if for every $x, y : X$ we have an equivalence $(x = y) \simeq (f(x) = f(y))$.*

Proof. See [Esc19b, Section 3.26]. ☐

Definition 2.4.7 (Subtype)**.** A *subtype* of a type X is a type A together with an embedding $A \hookrightarrow X$.

Lemma 2.4.8. *If Y is a proposition-valued type family over X, then $\Sigma_{x:X} Y(x)$ is a subtype of X, as witnessed by the first projection. In particular, two elements (x, y) and (x', y') of the Σ-type are equal if and only if $x = x'$.*

Proof. We have to show that all fibres of $\mathrm{pr}_1 : (\Sigma_{x:X} Y(x)) \to X$ are subsingletons. For arbitrary $x : X$ we have

$$\mathrm{fib}_{\mathrm{pr}_1}(x) \equiv \left(\Sigma_{(x',y):\Sigma Y}(x' = x)\right) \simeq (\Sigma_{x':X}(Y(x') \times (x' = x))) \simeq Y(x)$$

by reshuffling the Σ-types and the contractibility of the type $\Sigma_{x':X}(x' = x)$ at (x, refl). But $Y(x)$ is a subsingleton by assumption, proving that the first projection is an embedding. The second claim follows from Proposition 2.4.6. ☐

Remark 2.4.9. In the situation of Lemma 2.4.8 we often omit the second component of elements of $\Sigma_{x:X} Y(x)$ which is justified because any two elements of the second component are equal anyway.

Example 2.4.10. For any types X and Y, the types $X \hookrightarrow Y$ and $X \simeq Y$ are subtypes of $X \to Y$ by Theorem 2.4.3 and Lemma 2.4.8. Hence, two embeddings or equivalences are equal precisely when they are equal as ordinary maps. In particular, following Remark 2.4.9, we simply write $f : X \simeq Y$ for what is formally $(f, i) : X \simeq Y$ with i witnessing that f is an equivalence.

Definition 2.4.11 (Section, retraction and retract)**.** A *section* is a map $s : X \to Y$ together with a left inverse $r : Y \to X$, i.e. the maps satisfy $r(s(x)) = x$ for every $x : X$. We call r the *retraction* and say that X is a *retract* of Y.

Lemma 2.4.12. *Sections to sets are embeddings.*

Proof. Let $s : X \to Y$ be a section to a set. Since Y is a set, the type $s(x) = y$ is a proposition for every $x : X$ and $y : Y$. Hence, by Lemma 2.4.8, to prove that every fibre of s is a subsingleton, it suffices to prove that $x = x'$ whenever $s(x) = s(x')$. But if $s(x) = s(x')$, then we can apply the retraction on both sides to get $x = x'$. □

Remark 2.4.13. The restriction to sets in Lemma 2.4.12 is a necessary one, because [Shu16, Remark 3.11(2)] tells us that if every section is an embedding then every type is a set.

2.5 Function extensionality revisited

Armed with the notion of equivalence, we are now ready to give the official definition of function extensionality that improves on the naive version presented in Definition 2.2.6.

Definition 2.5.1 (**Axiom**: Function extensionality). *Function extensionality* asserts that for every two (dependent) functions $f, g : \Pi_{x:X} Y(x)$, the canonical map from $f = g$ to $\Pi_{x:X} f(x) = g(x)$ is an equivalence.

Naive function extensionality and function extensionality are logically equivalent, but the advantage of the above official formulation of function extensionality is that its type is a subsingleton, i.e. function extensionality is a property. So the distinction is similar to that between invertible maps and equivalences.

Proposition 2.5.2. *Assuming function extensionality, the type expressing function extensionality is a subsingleton.*

Proof. This follows from Theorems 2.4.3 and 2.3.17. □

2.6 Propositional truncation, images and surjections

We turn to introducing propositional truncations, which we will motivate through the problem of defining the image of a map. Intuitively, the image of $f : X \to Y$ should be the collection of elements $y : Y$ such that there exists $x : X$ with $f(x) = y$. A naive attempt at defining this might lead us to the type $\Sigma_{y:Y}\Sigma_{x:X}\, f(x) = y$.

Notice that Σ plays a double role here: the first Σ *collects* elements $y : Y$, while the second Σ supposedly expresses the existence of $x : X$ with $f(x) = y$. This hints at a problem and indeed there is one, because the type $\Sigma_{y:Y}\Sigma_{x:X}\, f(x) = y$ is obviously equivalent to $\Sigma_{x:X}\Sigma_{y:Y}\, f(x) = y$, which in turn is equivalent to just X, because for every $x : X$, the type $\Sigma_{y:Y}\, f(x) = y$ is a singleton by Theorem 2.3.5.

The above illustrates our lack of expressing "there exists" or "we have an *unspecified*". Instead of collecting $x : X$ for which $f(x) = y$, we only wish to record the knowledge that there is some $x : X$ with $f(x) = y$. We will do so by means of propositional truncations and subsequently use them to define images and surjections.

2.6.1 Propositional truncation

The propositional truncation is a higher inductive type; the only one we use in this thesis. We postulate a constructor $\|-\|$ that takes a type and returns a proposition: its *propositional truncation*. We require, unless explicitly stated otherwise, that our universes are closed under propositional truncations, i.e. if $X : \mathcal{U}$, then $\|X\| : \mathcal{U}$. Moreover, as part of the propositional truncation, we postulate that we have a map $|-| : X \to \|X\|$ for every type X.

We think of an element of $\|X\|$ as an *unspecified* element of X. The map $|-|$ then says that every specified elements gives rise to an unspecified one.

In the vocabulary of Martin-Löf Type Theory, the above gives a formation and an introduction rule for the propositional truncation, but we have not specified an elimination and computation rule yet. The elimination principle expresses that the propositional truncation is a reflector in the categorical sense: it is a left adjoint to the inclusion of proposition into all types. Spelled out it says that every map $f : X \to P$ to a proposition factors through $|-| : X \to \|X\|$, i.e. we have a map $\bar{f} : \|X\| \to P$ such that the diagram

$$\begin{tikzcd} X \arrow[rr, "f"] \arrow[dr, "|-|"'] & & P \\ & \|X\| \arrow[ur, dashed, "\bar{f}"'] & \end{tikzcd}$$

commutes.

What is paramount here is that this universal property holds for all propositions in arbitrary universes and not just for those in the same universe as X. We will return to this phenomenon in Section 2.11.2.

Some sources, e.g. [Uni13], also demand that the diagram above commutes *definitionally*: for every $x : X$, we have $f(x) \equiv \bar{f}(|x|)$. Having definitional equalities has some interesting consequences, such as being able to prove function extensionality [KECA17, Section 8]. We do not require definitional equalities, but notice that we do have $f(x) = \bar{f}(|x|)$ (up to an identification) for every $x : X$, as P is a subsingleton. In particular it follows using function extensionality that $\bar{f}$ is the unique factorisation.

From the universal property, we can prove that $\|-\|$ is a functor, because any map $f : X \to Y$ between types gives rise to a necessarily unique map $\|f\| : \|X\| \to \|Y\|$ such that $\|f\|(|x|) = |f(x)|$ for every $x : X$. Moreover, if a propositional truncation exists, then it is unique up to unique equivalence.

Definition 2.6.1 (Unspecified and specified existence, $\exists_{x:X} Y(x)$)**.**

(i) We suggestively write $\exists_{x:X} Y(x)$ for the propositional truncation of $\Sigma_{x:X} Y(x)$.

(ii) We say that "there exists (some) $x : X$ with $Y(x)$" or that we "have an unspecified $x : X$ with $Y(x)$" to mean that we have an element of $\exists_{x:X} Y(x)$.

(iii) By contrast, we say that we "have $x : X$ with $Y(x)$" or sometimes for emphasis, that we "have a *specified* $x : X$ with $Y(x)$" to mean that we have an element of $\Sigma_{x:X} Y(x)$.

The following lemma is sometimes known as “the type-theoretic axiom of choice”, which is a misnomer, because, as emphasised in the above distinction between specified and unspecified existence, there is no choice involved, since elements of Σ-types are specified witnesses. A correct formulation of the axiom of choice will be presented in Definition 2.7.24.

Lemma 2.6.2 (Distributivity of Π over Σ)**.** *For every type family Y over a type X and every family $P : \Pi_{x:X}(Y(x) \to \mathcal{U})$ we have an equivalence*

$$\left(\Pi_{x:X}\Sigma_{y:Y(x)}P(x,y)\right) \simeq \left(\Sigma_{f:\Pi Y}\Pi_{x:X}P(x,f(x))\right).$$

Proof. In the left-to-right direction, assume we have $\varphi : \Pi_{x:X}\Sigma_{y:Y(x)}P(x,y)$. Then, we define $f : \Pi Y$ by $f :\equiv \mathrm{pr}_1 \circ \varphi$ and we see that φ yields an element of $P(x, f(x))$ for every $x : X$. Conversely, if we have $f : \Pi Y$ and $\rho : \Pi_{x:X}P(x, f(x))$, then we define $\varphi : \Pi_{x:X}\Sigma_{y:Y(x)}P(x,y)$ as $\varphi(x) :\equiv (f(x), \rho(x))$. Finally, a direct computation shows that the maps above are indeed inverses. □

Definition 2.6.3 (Inhabited)**.** We say that a type is *inhabited* if we have an element of its propositional truncation.

Thus, a type is inhabited if we have an unspecified element of it. We do not use the word *nonempty* for this, because in our constructive setting this will mean something weaker, as explained in Section 2.7.

2.6.2 Images and surjections

As discussed, we now use the propositional truncation to define the image of map.

Definition 2.6.4 (Image, $\mathrm{im}(f)$, corestriction)**.** For a map $f : X \to Y$ we define

(i) its *image* as $\mathrm{im}(f) :\equiv \Sigma_{y:Y}\exists_{x:X}\, f(x) = y$, and
(ii) its *corestriction* as the map $f : X \to \mathrm{im}(f)$ given by $x \mapsto (f(x), |(x, \mathrm{refl})|)$.

Another take on the problem described in the introduction to this section is that the type $\Sigma_{y:Y}\Sigma_{x:X}\, f(x) = y$, being equivalent to X, is not a subtype of Y. Reassuringly, with the above official definition of image we do get a subtype.

Definition 2.6.5 (Surjection, $X \twoheadrightarrow Y$)**.** A map $f : X \to Y$ is a *surjection* if all fibres are inhabited. In others words, for every $y : Y$, there exists some $x : X$ with $f(x) = y$. The type of surjections from X to Y is denoted by $X \twoheadrightarrow Y$.

Lemma 2.6.6. *All corestrictions are surjections.*

Proof. By definition of the corestriction. □

Lemma 2.6.7 (Surjection induction)**.** *If $f : X \to Y$ is a surjection, then the following induction principle holds: for every subsingleton-valued $P : Y \to \mathcal{W}$, with $\mathcal{W}$ an* arbitrary *universe, if $P(f(x))$ holds for every $x : X$, then $P(y)$ holds for every $y : Y$.*

In the other direction, for any map $f : X \to Y$, if the above induction principle holds for the specific family $P(y) :\equiv \exists_{x:X}(f(x) = y)$, then f is a surjection.

Proof. Suppose that $f : X \to Y$ is a surjection, let $P : Y \to \mathcal{W}$ be subsingleton-valued and assume that $P(f(x))$ holds for every $x : X$. Now let $y : Y$ be arbitrary. We are to prove that $P(y)$ holds. Since f is a surjection, we have $\exists_{x:X}(f(x) = y)$. But $P(y)$ is a subsingleton, so, by the universal property of the propositional truncation, we may assume that we have a specific $x : X$ with $f(x) = y$. But then $P(y)$ must hold, because $P(f(x))$ does by assumption.
For the other direction, notice that if $P(y) :\equiv \exists_{x:X}(f(x) = y)$, then $P(f(x))$ clearly holds for every $x : X$. So by assuming that the induction principle applies, we get that $P(y)$ holds for every $y : Y$, which says exactly that f is a surjection. □

Lemma 2.6.8. *Every map $f : X \to Y$ factors as a surjection followed by an embedding: $X \twoheadrightarrow \mathrm{im}(f) \hookrightarrow Y$, where the first map is the corestriction of f and the second map is the first projection.*

Proof. That f is equal to the composite $X \twoheadrightarrow \mathrm{im}(f) \hookrightarrow Y$ is immediate. Moreover, the corestriction $X \to \mathrm{im}(f)$ is a surjection by Lemma 2.6.6, and the first projection $\mathrm{im}(f) \to Y$ is an embedding because of Lemma 2.4.8. □

2.6.3 Mapping from propositional truncations into sets

We recall a result due to Kraus, Escardó, Coquand and Altenkirch [KECA17] which has several applications throughout this thesis.

Theorem 2.6.9 ([KECA17, Theorem 5.4])**.** *Every constant map to a set factors through the truncation of its domain.*

Proof. Suppose that $f : X \to Y$ is a constant map to a set. By Lemma 2.6.8 and the universal property of the propositional truncation it suffices to prove that the image of f is a proposition, as this would yield the dashed map making the diagram

$$\begin{tikzcd}
& \|X\| \arrow[d, dashed] & \\
X \arrow[ur, "|{-}|"] \arrow[r, two heads] \arrow[rr, bend right, "f"'] & \mathrm{im}(f) \arrow[r, hook] & Y
\end{tikzcd}$$

commute. So suppose that we have $y, y' : Y$ such that there exists some $x : X$ with $f(x) = y$ and some $x' : X$ with $f(x') = y'$. By Lemma 2.4.8, we only have to prove that $y = y'$. But Y is assumed to be a set, so this a proposition. Hence, we can assume that we have specified $x : X$ and $x' : X$ with $f(x) = y$ and $f(x') = y'$. But then $y = f(x) = f(x') = y'$, as f is assumed to be constant. □

The theorem can be explained at a high level: since f is constant it does not matter what element we have of X (at least when Y is a set and has no higher dimensional structure). Thus, at least intuitively, as soon as we know that there exists some $x : X$ we should obtain a corresponding element of Y, because the choice of x is irrelevant.

2.7 Logic, (semi)decidability and constructivity

In univalent foundations and motivated by the discussion at the start of Section 2.6, we refine the Curry–Howard paradigm of propositions-as-types to propositions-as-subsingletons. That is, logical statements will be interpreted as types that have at most one element. For example, we interpret the existential quantifier as the propositional truncation of Σ. Thus, the logic in traditional set-level mathematics is encoded according to the following table.

Logical proposition	Subsingleton
True	$\mathbf{1}$
False	$\mathbf{0}$
P and Q	$P \times Q$
P implies Q	$P \to Q$
P or Q	$\lVert P + Q \rVert$
For every $x : X$ we have $P(x)$	$\Pi_{x:X} P(x)$
There exists $x : X$ such that $P(x)$	$\lVert \Sigma_{x:X} P(x) \rVert$

Table 2.7.1: Curry–Howard in univalent foundations.

Note that Theorems 2.3.16 and 2.3.17 ensure that $P \times Q$, $P \to Q$ and $\Pi_{x:X} P(x)$ are propositions if P, Q and each $P(x)$ are, while we use truncations to ensure that we get subsingletons for $\vee$ and the existential quantifier.

Definition 2.7.2 (Logical or, $\vee$)**.** We write $X \vee Y :\equiv \lVert X + Y \rVert$ for any two types X and Y.

Definition 2.7.3 (Negation, $\neg$)**.** The negation of a type X is denoted by $\neg X$ and defined as $X \to \mathbf{0}$.

Remark 2.7.4. Strictly speaking we should specify a universe for $\mathbf{0}$, but the choice is immaterial because $\mathbf{0}_{\mathcal{U}}$ and $\mathbf{0}_{\mathcal{V}}$ are easily seen to be equivalent for any two universes $\mathcal{U}$ and $\mathcal{V}$. For the sake of definiteness, we take $\mathbf{0}_{\mathcal{U}_0}$ in the definition of negation.

Note that the negation of any type is a subsingleton by Theorem 2.3.17 and the fact that $\mathbf{0}$ is a subsingleton.

Finally, it is important to be mindful of the fact that our logic will be constructive, as explained in Section 2.7.3.

2.7.1 Subsets and powersets

Definition 2.7.5 ($\mathcal{T}$-powerset, $\mathcal{T}$-valued subsets, $\mathcal{P}_{\mathcal{T}}(X)$, $x \in A$)**.** For any type X and a type universe $\mathcal{T}$, the *$\mathcal{T}$-powerset* $\mathcal{P}_{\mathcal{T}}(X)$ of X is defined as $X \to \Omega_{\mathcal{T}}$. We refer to its elements as *$\mathcal{T}$-valued subsets* of X. Given a $\mathcal{T}$-valued subset A of X and $x : X$, we write $x \in A$ for $A(x)$.

Lemma 2.7.6. *The $\mathcal{T}$-powerset of any type is a set.*

Proof. By Theorems 2.3.17 and 2.3.22. □

Definition 2.7.7 (Union, intersection, empty subset, singleton, $A \cup B$, $A \cap B$, $\emptyset$, $\{x\}$)**.** For $\mathcal{T}$-valued subsets A and B of a type X, we respectively write $A \cup B$ and $A \cap B$ for union and intersection that are formally defined by the maps $\lambda x \,.\, (x \in A) \vee (x \in B)$ and $\lambda x \,.\, (x \in A) \times (x \in B)$.
The empty subset of X is denoted by $\emptyset$ (leaving X and $\mathcal{T}$ implicit) and formally defined as $\lambda x \,.\, \mathbf{0}_{\mathcal{T}}$.
If X is a set in $\mathcal{T}$, then we have singleton subsets $\{x\}$ for every $x : X$, formally defined by $\lambda y \,.\, (x = y)$. Note that the requirement that X is a set in $\mathcal{T}$ is used to ensure that $x = y$ is indeed an element of $\Omega_{\mathcal{T}}$.

Of course, besides binary unions and intersections, we could use $\exists$ and Π to construct unions and intersections of families of subsets, but this matter is tightly connected to predicativity issues, so we will revisit it in some detail later.

Definition 2.7.8 (Subset inclusion, $\subseteq$)**.** Given a $\mathcal{T}$-valued subset A and a $\mathcal{W}$-valued subset B of a type $X : \mathcal{U}$, we write $A \subseteq B$ for the notion of *subset inclusion* that is formally defined as having an element of $\Pi_{x:X}(x \in A \to x \in B)$.

2.7.2 Decidability

In discussing constructive logic the notion of decidability is fundamental.

Definition 2.7.9 ((Weak) decidability of a type/equality)**.** A type X

(i) is *decidable* if we have an element of $X + \neg X$,
(ii) is *weakly decidable* if we have an element of $\neg\neg X + \neg X$, and
(iii) has *decidable equality* if $x = y$ is decidable for every $x, y : X$.

Note that (i) is data, while (ii) and (iii) are property. For (ii), this holds because negated types are subsingletons and because $\neg\neg X$ and $\neg X$ are mutually exclusive. For (iii), this is a consequence of Hedberg's Theorem, which is Theorem 2.7.12 below.

Example 2.7.10. The types $\mathbf{0}$, $\mathbf{1}$ and $\mathbb{N}$ all have decidable equality.

Lemma 2.7.11. *If we have maps back and forth between two types X and Y and one of the types is decidable, then so is the other.*

Proof. Suppose that $f\colon X \to Y$ and $g\colon Y \to X$ and that X is decidable. Then we have $x : X$ or $\neg X$. In the first case we get $f(x) : Y$ and in the second case, we get $\neg Y$ by contraposition and $g : Y \to X$. □

Theorem 2.7.12 (Hedberg's Theorem [Hed98])**.** *If a type has decidable equality, then it is a set.*

Proof. Suppose that X is a type with decidable equality. By Hedberg's Lemma (Lemma 2.3.13), it suffices to construct a constant endofunction on $x = y$ for every $x, y : X$. By assumption, we either have $p : x = y$ or $x \neq y$. In the latter case, $x = y$ is equivalent to $\mathbf{0}$ and so it certainly has a constant endomap. And if we have $p : x = y$, then the function $(x = y) \to (x = y)$ mapping everything to p is constant. □

When studying the Scott model of PCF in Section 5.2, the notion of *semi*decidability also makes an appearance. Intuitively, a proposition is semidecidable if it can be affirmed through some finite procedure. Notice how we do not impose such a restriction on it being refuted, so this notion is characteristically asymmetric.

Definition 2.7.13 (Semidecidability)**.** A proposition P is *semidecidable* if there exists some binary sequence $\alpha : \mathbb{N} \to \mathbf{2}$ such that P holds if and only if there exists some $n : \mathbb{N}$ for which $\alpha(n) = 1$.

Indeed, if a semidecidable proposition P holds, then we will eventually find $n : \mathbb{N}$ for which $\alpha(n)$. But inspecting the sequence α for any finite number of values will never allow us to conclude that the negation of P holds as this would require knowing that the sequence *never* attains the value 1. Also notice that every decidable proposition is semidecidable.

Lemma 2.7.14. *A proposition P is semidecidable if and only if there exists a natural number $k : \mathbb{N}$ and a family $Q : \mathbb{N}^k \to \mathcal{U}_0$ such that $Q(\vec{n})$ is decidable for all inputs $\vec{n} : \mathbb{N}^k$ and P holds exactly when $\exists_{\vec{n}:\mathbb{N}^k} Q(\vec{n})$ does.*

Proof. Note that the type of decidable propositions is equivalent to $\mathbf{2}$ and that for every natural number k we have a bijection $\mathbb{N}^k \simeq \mathbb{N}$, so that we can always turn a family with k parameters into a corresponding one with a single parameter. □

2.7.3 Constructivity

Our foundational setup will be constructive in the sense that we do not assume any additional logical axioms beyond propositional extensionality and function extensionality. In particular, we do not assume (weak) excluded middle (Definition 2.7.19), the limited principle of omniscience (LPO; Definition 2.7.22) or the axiom of (countable) choice (Definition 2.7.24), as these are constructively unacceptable [Bis67, p. 9], and even provably false in some varieties of constructive mathematics [BR87, pp. 3–4]. In MLTT, and univalent foundations, they are simply independent: they cannot be proved, but neither can their negations.

This does *not* mean that these logical principles will have no use for us. In fact, they will feature as *constructive taboos*. That is, sometimes we wish to argue that something is inherently nonconstructive and we can do so as follows: if we can show that X implies excluded middle, then this tells us that X is constructively unacceptable, because excluded middle is.

Definition 2.7.15 (Nonemptiness)**.** A type X is *nonempty* if $\neg\neg X$ has an element.

Lemma 2.7.16. *Every inhabited type is nonempty.*

Proof. Let X be an inhabited type, i.e. we have an element t of $\|X\|$. Since being nonempty is a proposition by Theorem 2.3.17, we can eliminate t and assume to have $x : X$. But then $\lambda\,(f : \neg X)\,.\,f(x)$ is an element of $\neg\neg X$. □

Definition 2.7.17 ($\neg\neg$-stability, type of $\neg\neg$-stable propositions, $\Omega_{\mathcal{U}}^{\neg\neg}$)**.**
- (i) A type X is said to be *$\neg\neg$-stable* if we have an element of $\neg\neg X \to X$.
- (ii) We denote the *type of $\neg\neg$-stable propositions* in a universe $\mathcal{U}$ by $\Omega_{\mathcal{U}}^{\neg\neg}$.

Remark 2.7.18. Using the terminology of Definition 2.6.3 and the observation that a proposition is equivalent to its propositional truncation, we see that a proposition is $\neg\neg$-stable precisely when it is inhabited as soon as it is nonempty.

Definition 2.7.19 ((Weak) excluded middle)**.** For a type universe $\mathcal{U}$, *(weak) excluded middle* in $\mathcal{U}$ asserts that every proposition in $\mathcal{U}$ is (weakly) decidable.

Remark 2.7.20. The restriction to propositions in the formulation of (weak) excluded middle can be explained in two ways. Firstly, given our interpretation of (logical)-propositions-as-subsingletons, it seems appropriate to restrict (weak) excluded middle to the logical fragment of our framework. In fact, the statement $X + \neg X$ for all types X is *global choice*: it says that we can choose a specified element of every nonempty type, and is incompatible with univalence [Esc19b, Section 3.35.6]. Secondly, the unrestricted formulation is provably false in the presence of the univalence axiom, while the restricted formulation is consistent with univalent type theory, as shown by Voevodsky's simplicial sets model [KL21].

Lemma 2.7.21. *The following logical statements are equivalent for a universe $\mathcal{U}$:*
- *(i) excluded middle in $\mathcal{U}$;*
- *(ii) every proposition in $\mathcal{U}$ is equal to either $\mathbf{0}_{\mathcal{U}}$ or $\mathbf{1}_{\mathcal{U}}$;*
- *(iii) the type $\Omega_{\mathcal{U}}$ has decidable equality;*
- *(iv) the map $2_{\mathcal{V}} \to \Omega_{\mathcal{U}}$ given by $0 \mapsto \mathbf{0}_{\mathcal{U}}$ and $1 \mapsto \mathbf{1}_{\mathcal{U}}$ is an equivalence for any type universe $\mathcal{V}$;*
- *(v) all elements of $\Omega_{\mathcal{U}}$ are $\neg\neg$-stable* (double negation elimination)*;*
- *(vi) every nonempty type is inhabited.*

Similarly, the following logical statements are equivalent for every universe $\mathcal{U}$:
- *(i') weak excluded middle in $\mathcal{U}$;*

(ii') *every* $\neg\neg$*-stable proposition in* $\mathcal{U}$ *is equal to either* $\mathbf{0}_{\mathcal{U}}$ *or* $\mathbf{1}_{\mathcal{U}}$*;*
(iii') *the type* $\Omega_{\mathcal{U}}^{\neg\neg}$ *has decidable equality;*
(iv') *the map* $\mathbf{2}_{\mathcal{V}} \to \Omega_{\mathcal{U}}^{\neg\neg}$ *given by* $0 \mapsto \mathbf{0}_{\mathcal{U}}$ *and* $1 \mapsto \mathbf{1}_{\mathcal{U}}$ *is an equivalence for any type universe* $\mathcal{V}$*.*

Proof. For the equivalence of (i)–(iii), observe that, by propositional extensionality, a proposition holds if and only if it is equal to the unit type and does not hold if and only if it is equal to the empty type. It is also clear that (iv) implies (ii). For the converse, assume (ii) and note that it implies that

$$\Omega_{\mathcal{U}} \simeq \Sigma_{P:\mathcal{U}}((P = \mathbf{1}_{\mathcal{U}}) + (P = \mathbf{0}_{\mathcal{U}})) \simeq (\Sigma_{P:\mathcal{U}}\, P = \mathbf{1}_{\mathcal{U}}) + (\Sigma_{P:\mathcal{U}}\, P = \mathbf{0}_{\mathcal{U}}) \simeq \mathbf{2}_{\mathcal{V}},$$

where the final equivalence holds by Theorem 2.3.5 and the fact that every contractible type is equivalent to $\mathbf{1}_{\mathcal{V}}$.
The equivalence of (i) and (v) is well-known in constructive logic. If we assume excluded middle in $\mathcal{U}$ and P is an arbitrary proposition in $\mathcal{U}$, then we have P or $\neg P$. In the first case, obviously $\neg\neg P \to P$ and in the second case this also holds, because the antecedent of the implication contradicts $\neg P$. Conversely, if double negation elimination for $\mathcal{U}$ holds and P is an arbitrary proposition in $\mathcal{U}$, then in particular the proposition $P + \neg P$ is $\neg\neg$-stable. But $\neg\neg(P + \neg P)$ is a tautology: for if we assume $\neg(P + \neg P)$, then assuming either P or $\neg P$ would yield a contradiction, hence we have $\neg P \times \neg\neg P$, which is false. Thus, by our double negation elimination assumption, we get decidability of P, completing the proof that items (i)–(v) are equivalent.
The equivalence of (v) and (vi) follows from the observations that every proposition is equivalent to its propositional truncation, and that $\neg\neg X$ is equivalent to $\neg\neg\|X\|$.
The final claim of the lemma, concerning (i')–(iv'), follows from the fact that decidability and weak decidability coincide for $\neg\neg$-stable propositions. □

Definition 2.7.22 (Limited principle of omniscience (LPO))**.** Bishop's *limited principle of omniscience (LPO)* asserts that for every binary sequence $\alpha : \mathbb{N} \to \mathbf{2}$ the proposition $\exists_{n:\mathbb{N}}\, \alpha(n) = 1$ is decidable.

Remark 2.7.23. Unfolding the definitions, we see that LPO says exactly that every semidecidable proposition is decidable.

Definition 2.7.24 (Axiom of (countable) choice)**.** The *axiom of choice* with respect to universes $\mathcal{U}$ and $\mathcal{V}$ says that for every set $X : \mathcal{U}$ and set-valued type-family $Y : X \to \mathcal{V}$, if every $Y(x)$ is inhabited, then $\Pi_{x:X} Y(x)$ is inhabited as well. Symbolically, this reads

$$(\Pi_{x:X}\|Y(x)\|) \to \|\Pi_{x:X} Y(x)\| \tag{2.7.25}$$

The special case where $X \equiv \mathbb{N}$ is called *countable choice.*

There are multiple equivalent ways of phrasing the axiom of choice in univalent foundations, see [Esc19b, Section 3.35], but the above is the most convenient formulation for us.

Remark 2.7.26. Semidecidability and countable choice are closely linked as investigated in [Kna18; EK17] and further in [dJon22c], but it is somewhat beyond the scope of this thesis to go into this here.

2.8 Univalent universes

By analogy to propositional extensionality (Definition 2.3.20) and function extensionality (Definition 2.5.1), we define an extensionality axiom for *types* and say that a universe is univalent if its types satisfy it.

Definition 2.8.1 (Univalence). A type universe $\mathcal{U}$ is *univalent* if for every $X, Y : \mathcal{U}$ the map $(X =_{\mathcal{U}} Y) \to (X \simeq Y)$ defined by path induction as $\mathsf{refl} \mapsto \mathsf{id}$ is an equivalence.

In other words, two types are equal precisely when they are equivalent, although the formulation above is carefully chosen to ensure (through Theorems 2.4.3 and 2.3.17) that being univalent is a property of a universe.

The *univalence axiom* asserts that all universes are univalent. Unlike propositional extensionality and function extensionality, we do not assume this globally, but rather add the univalence of a universe as an explicit hypothesis to our theorems when needed.

The consistency of the univalence axiom was established by Voevodsky through the simplicial sets model [KL21].

Theorem 2.8.2. *If $\mathcal{U}$ is univalent, then we have propositional extensionality in $\mathcal{U}$ and function extensionality for functions between types in $\mathcal{U}$.*

Proof. That univalence implies propositional extensionality is straightforward as two propositions are equivalent precisely when they imply each other. That function extensionality can be derived from univalence is due to Voevodsky, see [Esc19b, Section 3.17] or [Uni13, Section 4.9] for proofs. □

The following result should be compared to Theorem 2.3.5.

Theorem 2.8.3. *A universe $\mathcal{U}$ is univalent if and only if for every $X : \mathcal{U}$ the type $\Sigma_{Y:\mathcal{U}} X \simeq Y$ is contractible.*

Proof. See [Esc19b, Section 3.14]. □

Definition 2.8.4 (Universe embedding; [Esc19b, Section 3.30]). A function f between universes $\mathcal{U}$ and $\mathcal{V}$ is a *universe embedding* if $f(X) \simeq X$ for every $X : \mathcal{U}$.

Proposition 2.8.5 ([Esc19b, Section 3.30]). *If $\mathcal{U}$ and $\mathcal{V}$ are univalent universes, then every universe embedding between $\mathcal{U}$ and $\mathcal{V}$ is an embedding.*

Proof. Suppose that $\mathcal{U}$ and $\mathcal{V}$ are univalent and that f is a universe embedding. Then for every $X, Y : \mathcal{U}$, we have

$$\begin{aligned}(X = Y) &\simeq (X \simeq Y) && \text{(since } \mathcal{U} \text{ is univalent)}\\ &\simeq (f(X) \simeq f(Y)) && \text{(since } f \text{ is a universe embedding)}\\ &\simeq (f(X) = f(Y)) && \text{(since } \mathcal{V} \text{ is univalent)}\end{aligned}$$

so that f is an embedding by Proposition 2.4.6. □

Corollary 2.8.6. *If the universes $\mathcal{U}$ and $\mathcal{U} \sqcup \mathcal{V}$ are univalent, then the universe embedding* $\mathrm{lift}_{\mathcal{U},\mathcal{V}} : \mathcal{U} \to \mathcal{U} \sqcup \mathcal{V}$ *defined by $X \mapsto X \times \mathbf{1}_{\mathcal{V}}$ is an embedding.*

2.9 Small and locally small types

A fundamental theme of this work will be the concept of a (locally) small type, as (im)predicativity is all about whether (types of) propositions are small or not.

Definition 2.9.1 ((Local) $\mathcal{U}$-smallness; Rijke [Rij17])**.** A type X in any universe is

(i) *$\mathcal{U}$-small* if it is equivalent to a type in the universe $\mathcal{U}$, i.e.

$$X \text{ is } \mathcal{U}\text{-small} :\equiv \textstyle\sum_{Y:\mathcal{U}} (Y \simeq X),$$

(ii) *locally $\mathcal{U}$-small* if the type $(x = y)$ is $\mathcal{U}$-small for every $x, y : X$.

From a categorical perspective, $\mathcal{U}$-small really means *essentially* $\mathcal{U}$-small, because we are considering types up to equivalence. We simply call it $\mathcal{U}$-smallness, because the corresponding strict notions where Y is equal (definitional, or up to the intensional identity type) to X can only make sense if X is already in the same universe as Y.

A fact that we will often use tacitly is the useful but simple observation that if X is $\mathcal{U}$-small and $X \simeq Y$ then Y is $\mathcal{U}$-small too.

Example 2.9.2.

(i) Every $\mathcal{U}$-small type is locally $\mathcal{U}$-small.

(ii) The type $\Omega_{\mathcal{U}}$ of propositions in a universe $\mathcal{U}$ lives in $\mathcal{U}^+$, but is locally $\mathcal{U}$-small by propositional extensionality.

Even though we phrased $\mathcal{U}$-smallness using equivalences, a type can be $\mathcal{U}$-small in at most one way, provided that the universes involved are univalent.

Proposition 2.9.3. *If $\mathcal{V}$ and $\mathcal{U} \sqcup \mathcal{V}$ are univalent universes, then the type expressing that X is $\mathcal{V}$-small is a proposition for every $X : \mathcal{U}$.*

Proof. If $\mathcal{U} \sqcup \mathcal{V}$ is univalent, then

$$\begin{aligned} X \text{ is } \mathcal{V}\text{-small} &\equiv \Sigma_{Y:\mathcal{V}}(Y \simeq X) \\ &\simeq \Sigma_{Y:\mathcal{V}}(\operatorname{lift}_{\mathcal{V},\mathcal{U}}(Y) \simeq \operatorname{lift}_{\mathcal{U},\mathcal{V}}(X)) \quad \text{(as the lifts are universe embeddings)} \\ &\simeq \Sigma_{Y:\mathcal{V}}(\operatorname{lift}_{\mathcal{V},\mathcal{U}}(Y) = \operatorname{lift}_{\mathcal{U},\mathcal{V}}(X)) \quad \text{(as } \mathcal{U} \sqcup \mathcal{V} \text{ is univalent)} \\ &\equiv \operatorname{fib}_{\operatorname{lift}_{\mathcal{V},\mathcal{U}}}(\operatorname{lift}_{\mathcal{U},\mathcal{V}}(X)). \end{aligned}$$

But the latter is a proposition because $\operatorname{lift}_{\mathcal{V},\mathcal{U}}$ is an embedding by Corollary 2.8.6 using our assumptions that $\mathcal{V}$ and $\mathcal{U} \sqcup \mathcal{V}$ are univalent. □

The converse also holds in the following form.

Proposition 2.9.4. *The type expressing that X is $\mathcal{U}$-small is a proposition for every $X : \mathcal{U}$ if and only if the universe $\mathcal{U}$ is univalent.*

Proof. This follows from Theorem 2.8.3. □

Lemma 2.9.5. *Propositional extensionality suffices to prove that being $\mathcal{U}$-small is a property for propositions.*

Proof. For propositions we see that the argument of Proposition 2.9.3 and its dependency Proposition 2.8.5 only require propositional extensionality and not full univalence. □

We end this section by showing our main technical result on small types here, namely that being small is closed under retracts. The following original definition extends the notion of a small type to functions.

Definition 2.9.6 ($\mathcal{U}$-smallness for maps)**.** A map $f : X \to Y$ is said be $\mathcal{U}$-small if all its fibres are.

Lemma 2.9.7.

(i) *A type X is $\mathcal{U}$-small if and only if the unique map $X \to \mathbf{1}_{\mathcal{U}_0}$ is $\mathcal{U}$-small.*
(ii) *If Y is $\mathcal{U}$-small, then a map $f : X \to Y$ is $\mathcal{U}$-small if and only if X is.*

Proof. (i) Writing $!_X$ for the map $X \to \mathbf{1}_{\mathcal{U}_0}$ we have $\operatorname{fib}_{!_X}(\star) \simeq X$. (ii) If X and Y are both $\mathcal{U}$-small, witnessed respectively by $\varphi : X' \simeq X$ and $\psi : Y' \simeq Y$, then $\operatorname{fib}_f(y)$ is $\mathcal{U}$-small for every $y : Y$, because $\operatorname{fib}_f(y) \equiv \Sigma_{x:X}(f(x) = y) \simeq \Sigma_{x':X'}(\psi^{-1}(f(\varphi(x'))) = \psi^{-1}(y))$. Conversely, if f and Y are $\mathcal{U}$-small, then so is X, because [Uni13, Lemma 4.8.2] tells us that $X \simeq \Sigma_{y:Y} \operatorname{fib}_f(y)$. □

Theorem 2.9.8. *Every section into a $\mathcal{U}$-small type is $\mathcal{U}$-small. In particular, its domain is $\mathcal{U}$-small.*

Proof. We show that the domain is $\mathcal{U}$-small from which it follows that the section is $\mathcal{U}$-small by Lemma 2.9.7(ii). So suppose we have a section $s : X \to Y$ with retraction

$r : Y \to X$ and that Y is $\mathcal{U}$-small. By [Shu16, Lemma 3.6], the endomap $f :\equiv r \circ s$ on Y is a quasi-idempotent [Shu16, Definition 3.5]. Hence, [Shu16, Theorem 5.3] tells us that f can be split as $Y \xrightarrow{r'} A \xrightarrow{s'} Y$ for some maps s' and r' and some type A recalled below. Now X and A are equivalent as witnessed by the maps $x \mapsto r'(s(x))$ and $a \mapsto r(s'(a))$. Finally, we recall from the proof of [Shu16, Theorem 5.3] that $A :\equiv \Sigma_{\sigma:\mathbb{N}\to Y}\Pi_{n:\mathbb{N}}(f(\sigma_{n+1}) = \sigma_n)$ which is $\mathcal{U}$-small because Y is assumed to be. □

Remark 2.9.9. In [dJE21b] we had a weaker version of Theorem 2.9.8 where we included the additional assumption that the section was an embedding. (Note that if every section is an embedding, then every type is a set [Shu16, Remark 3.11(2)], but that all sections into *sets* are embeddings [Esc19b, `lc-maps-into-sets-are-embeddings`].) We are grateful to the anonymous reviewer of [dJE22a] who proposed the above strengthening.

2.10 Impredicativity: resizing axioms

We have already explained in Section 2.7.3 that our setup is constructive because we do not assume excluded middle or the axiom of choice. Similarly, our setup is *predicative* because we do not assume certain resizing principles concerning propositions which we define below. Recall that the type of all propositions in a universe $\mathcal{U}$ is denoted by $\Omega_{\mathcal{U}}$ and lives in $\mathcal{U}^+$. Similarly, the type of all $\neg\neg$-stable propositions in $\mathcal{U}$ is denoted by $\Omega_{\mathcal{U}}^{\neg\neg}$ and also lives in $\mathcal{U}^+$.

Definition 2.10.1 (Propositional resizing)**.**

(i) By *Propositional-Resizing*$_{\mathcal{U},\mathcal{V}}$ we mean the assertion that every proposition P in a universe $\mathcal{U}$ is $\mathcal{V}$-small.
(ii) We write Ω*-Resizing*$_{\mathcal{U},\mathcal{V}}$ for the assertion that the type $\Omega_{\mathcal{U}}$ is $\mathcal{V}$-small.
(iii) By $\Omega_{\neg\neg}$*-Resizing*$_{\mathcal{U},\mathcal{V}}$ we mean the assertion that the type $\Omega_{\mathcal{U}}^{\neg\neg}$ is $\mathcal{V}$-small.
(iv) For the particular case of a single universe, we respectively write Ω*-Resizing*$_{\mathcal{U}}$ and $\Omega_{\neg\neg}$*-Resizing*$_{\mathcal{U}}$ for the assertions that $\Omega_{\mathcal{U}}$ is $\mathcal{U}$-small and $\Omega_{\mathcal{U}}^{\neg\neg}$ is $\mathcal{U}$-small.

The resizing of the type of propositions in a universe (ii) is closely related to the resizing of the propositions themselves (i), as we show now.

Proposition 2.10.2 (cf. [Esc19b, Section 3.36.4])**.** *For every two type universes $\mathcal{U}$ and $\mathcal{V}$ we have that*

(i) Ω-Resizing$_{\mathcal{U},\mathcal{V}}$ *implies* Propositional-Resizing$_{\mathcal{U},\mathcal{V}}$*, and*
(ii) the conjunction of Propositional-Resizing$_{\mathcal{U},\mathcal{V}}$ *and* Propositional-Resizing$_{\mathcal{V},\mathcal{U}}$ *implies* Ω-Resizing$_{\mathcal{U},\mathcal{V}^+}$.

Proof. (i) Assuming Ω-Resizing$_{\mathcal{U},\mathcal{V}}$ we have an equivalence $\varphi : \Omega_{\mathcal{U}} \simeq T : \mathcal{V}$, so if P is an arbitrary proposition in $\mathcal{U}$, then P holds if and only if $P = \mathbf{1}_{\mathcal{U}}$ if and only if $\varphi(P) = \varphi(\mathbf{1}_{\mathcal{U}})$, but the latter is a type in $\mathcal{V}$. (ii) Assuming Propositional-Resizing$_{\mathcal{U},\mathcal{V}}$ and Propositional-Resizing$_{\mathcal{V},\mathcal{U}}$ yields maps $\phi : \Omega_{\mathcal{U}} \to \Omega_{\mathcal{V}}$ and $\psi : \Omega_{\mathcal{V}} \to \Omega_{\mathcal{U}}$ such that for every $P : \Omega_{\mathcal{U}}$ and $Q : \Omega_{\mathcal{V}}$ we have $P \simeq \phi(P)$ and $Q \simeq \psi(Q)$. Hence, for every $P : \Omega_{\mathcal{U}}$ we have $P \simeq \phi(P) \simeq \psi(\phi(P))$ and similarly $Q \simeq \phi(\psi(Q))$ for every $Q : \Omega_{\mathcal{V}}$. Thus, by propositional extensionality, we get an equivalence $\Omega_{\mathcal{U}} \simeq \Omega_{\mathcal{V}}$, but $\Omega_{\mathcal{V}} : \mathcal{V}^+$, so Ω-Resizing$_{\mathcal{U},\mathcal{V}^+}$ must hold. □

Remark 2.10.3. In light of the occurrence of $\mathcal{V}^+$ in Proposition 2.10.2(ii), it is worth observing that if we assume the strong principle Propositional-Resizing$_{\mathcal{U},\mathcal{U}_0}$ then Ω-Resizing$_{\mathcal{U},\mathcal{U}_1}$ holds. Hence, if we can resize every proposition to one in the lowest universe $\mathcal{U}_0$, then the type of propositions in an arbitrary universe is equivalent to a type in $\mathcal{U}_1$, which is a significant resizing for all universes except $\mathcal{U}_0$.

With the classical axiom of excluded middle, impredicativity becomes a theorem. Thus, if we wish to explore predicativity (in the form of propositional resizing axioms) in univalent foundations then we must work constructively.

Proposition 2.10.4 (cf. [Esc19b, Section 3.36.2]). *Excluded middle implies impredicativity. Specifically,*

(i) *excluded middle in $\mathcal{U}$ implies Ω-Resizing$_{\mathcal{U},\mathcal{U}_0}$, and*
(ii) *weak excluded middle in $\mathcal{U}$ implies $\Omega_{\neg\neg}$-Resizing$_{\mathcal{U},\mathcal{U}_0}$.*

Proof. (i) By Lemma 2.7.21(iv) we know that excluded middle in $\mathcal{U}$ implies having an equivalence $\Omega_{\mathcal{U}} \simeq \mathbf{2}_{\mathcal{U}_0}$. (ii) By Lemma 2.7.21(iv'). □

2.11 Quotients, replacement, and propositional truncations revisited

We investigate the inter-definability and interaction of type universe levels of propositional truncations and set quotients in the absence of propositional resizing axioms. In particular, we will see that it is not so important if the set quotient or propositional truncation lives in a higher universe. What is paramount instead is whether the universal property applies to types in arbitrary universes. However, in some cases, like in Section 6.3.2, it is relevant whether set quotients are small and we show this to be equivalent to a set replacement principle in Section 2.11.4.

Remark 2.11.1. Recall from Section 2.6.1 that in this thesis we typically assume our universes to be closed under propositional truncations. However, in this section, we will be more general and assume that $\|X\| : F(\mathcal{U})$ where F is a metafunction on universes, so that the above case is obtained by taking F to be the identity. We will also consider $F(\mathcal{U}) = \mathcal{U}_1 \sqcup \mathcal{U}$ in the final subsection.

2.11.1 Propositional truncations and propositional resizing

Voevodsky [Voe11] introduced propositional resizing rules in order to construct propositional truncations [PVW15, Section 2.4]. Here we review Voevodsky's construction, paying special attention to the universes involved.

NB. We do not assume the availability of propositional truncations in this section.

Definition 2.11.2 (Voevodsky propositional truncation, $\|X\|_v$)**.** The *Voevodsky propositional truncation* $\|X\|_v$ of a type $X : \mathcal{U}$ is defined as

$$\|X\|_v :\equiv \prod_{P:\mathcal{U}} (\text{is-subsingleton}(P) \to (X \to P) \to P).$$

Observe that this is a System-F [Gir71; Rey74; AFS18] style definition where we use the desired universal property and a (large) quantification to encode a type.

Also notice that the part $(X \to P) \to P$ in the Voevodsky propositional truncation generalises the double negation (which is given by taking $P \equiv \mathbf{0}$). The double negation of a type is a proposition, but it enjoys the universal property of the propositional truncation if and only if excluded middle holds [KECA17, Section 7].

Because of Theorem 2.3.17, one can show that $\|X\|_v$ is indeed a proposition for every type X. Moreover, we have a map $|-|_v : X \to \|X\|_v$ given by $|x|_v :\equiv (P, i, f) \mapsto f(x)$.

Observe that $\|X\|_v : \mathcal{U}^+$, so using the notation from Remark 2.11.1, we have $F(\mathcal{U}) :\equiv \mathcal{U}^+$. However, as we will argue for set quotients, it does not matter so much where the truncated proposition lives; it is much more important that we can eliminate into subsingletons in arbitrary universes, i.e. that $\|-\|_v$ satisfies the right universal property. Given $X : \mathcal{U}$ and a map $f : X \to P$ to a proposition $P : \mathcal{U}$ with $i : \text{is-subsingleton}(P)$, we have a map $\|X\|_v \to P$ given as $\Phi \mapsto \Phi(P, i, f)$. However, if the proposition P lives in some other universe $\mathcal{V}$, then we seem to be completely stuck. To clarify this, we consider the example of functoriality.

Example 2.11.3. If we have a map $f : X \to Y$ with $X : \mathcal{U}$ and $Y : \mathcal{U}$, then we get a lifting simply by precomposition, i.e. we define $|f|_v : \|X\|_v \to \|Y\|_v$ by $|f|_v(\Phi) :\equiv (P, i, g) \mapsto \Phi(P, i, g \circ f)$. But obviously, we also want functoriality for maps $f : X \to Y$ with $X : \mathcal{U}$ and $Y : \mathcal{V}$, but this is impossible with the above definition of $|f|_v$, because for $\|X\|_v$ we are considering propositions in $\mathcal{U}$, while for $\|Y\|_v$ we are considering propositions in $\mathcal{V}$.
In particular, even if the types $X : \mathcal{U}$ and $Y : \mathcal{V}$ are equivalent, then it does not seem possible to construct an equivalence between $\|X\|_v$ and $\|Y\|_v$. This issue also comes up if one tries to prove that the map $|-|_v : X \to \|X\|_v$ is a surjection [Esc19b, Section 3.34.1].

Proposition 2.11.4 (cf. [KECA17, Theorem 3.8])**.** *If our type theory has propositional truncations with $\|X\| : \mathcal{U}$ whenever $X : \mathcal{U}$, then $\|X\|_v$ is $\mathcal{U}$-small.*

Proof. We will show that $\|X\|$ and $\|X\|_v$ are logically equivalent (i.e. we have maps in both directions), which suffices, because both types are subsingletons. We obtain a map $\|X\| \to \|X\|_v$ by applying the universal property of $\|X\|$ to the map $|-|_v : X \to \|X\|_v$. Observe that it is essential that the universal property allows for elimination into subsingletons in universes other than $\mathcal{U}$, as $\|X\|_v : \mathcal{U}^+$. For the function in the other direction, simply note that $\|X\| : \mathcal{U}$, so that we can construct $\|X\|_v \to \|X\|$ as $\Phi \mapsto \Phi(\|X\|, i, |-|)$ where i witnesses that $\|X\|$ is a subsingleton. □

Thus, as is folklore in the univalent foundations community, we can view higher inductive types as specific resizing axioms. But note that the converse to the above proposition does not appear to hold, because even if $\|X\|_v$ is $\mathcal{U}$-small, then it still wouldn't have the appropriate universal property. This is because the definition of $\|X\|_v$ is a dependent product over propositions in $\mathcal{U}$ only, which now includes $\|X\|_v$, but still misses propositions in other universes. In the presence of resizing axioms, we could obtain the full universal property, because we would have (equivalent copies of) all propositions in a single universe:

Proposition 2.11.5 (cf. [Esc19b, Section 36.5])**.** *If* Propositional-Resizing$_{\mathcal{U},\mathcal{U}_0}$ *holds for every universe $\mathcal{U}$, then the Voevodsky propositional truncation satisfies the full universal property with respect to all types in all universes.*

2.11.2 Set quotients from propositional truncations

In this section we assume to have propositional truncations with $\|X\| : F(\mathcal{U})$ when $X : \mathcal{U}$ for some metafunction F on universes. We will be mainly interested in $F(\mathcal{U}) = \mathcal{U}$ and $F(\mathcal{U}) = \mathcal{U}_1 \sqcup \mathcal{U}$ for the reasons explained below.

We prove that we can construct set quotients using propositional truncations. The construction is due to Voevodsky and also appears in [Uni13, Section 6.10] and [RS15, Section 3.4]. While Voevodsky assumed propositional resizing rules in his construction, we show, following [Esc19b, Section 3.37], that resizing is not needed to prove the universal property of the set quotient, provided propositional truncations are available.

Definition 2.11.6 (Equivalence relation)**.** An *equivalence relation* on a type X is a binary type family $\approx : X \to X \to \mathcal{V}$ that is

(i) subsingleton-valued, i.e. $x \approx y$ is a subsingleton for every $x, y : X$,
(ii) reflexive, i.e. $x \approx x$ for every $x : X$,
(iii) symmetric, i.e. $x \approx y$ implies $y \approx x$ for every $x, y : X$, and
(iv) transitive, i.e. the conjunction of $x \approx y$ and $y \approx z$ implies $x \approx z$ for every $x, y, z : X$.

Definition 2.11.7 (Set quotient, $X/{\approx}$)**.** We define the *set quotient* $X/{\approx}$ of X by $\approx$ to be the image of $e_\approx$ where

$$\begin{aligned} e_\approx &: X \to (X \to \Omega_{\mathcal{V}}) \\ &\quad x \mapsto (y \mapsto (x \approx y, p(x, y))) \end{aligned}$$

and p is the witness that $\approx$ is subsingleton-valued.

Of course, we should prove that $X/{\approx}$ really is the quotient of X by $\approx$ by proving a suitable universal property. The following definition and lemmas indeed build up to this. For the remainder of this section, we will fix a type $X : \mathcal{U}$ with an equivalence relation $\approx : X \to X \to \mathcal{V}$.

Remark 2.11.8. Since $X/{\approx} \equiv \mathrm{im}(e_{\approx}) \equiv \Sigma_{\varphi:X\to\Omega_{\mathcal{V}}}\exists_{x:X}(\lambda\, y\, .\, x \approx y) = \varphi$, we see, recalling Remark 2.11.1 and the fact that $\Omega_{\mathcal{V}}$ is a type in $\mathcal{V}^+$, that we have $X/{\approx} : \mathcal{T} \sqcup F(\mathcal{T})$ with $\mathcal{T} :\equiv \mathcal{V}^+ \sqcup \mathcal{U}$. In the particular case that F is the identity, we obtain the simpler $X/{\approx} : \mathcal{V}^+ \sqcup \mathcal{U}$.

Lemma 2.11.9. *The quotient $X/{\approx}$ is a set.*

Proof. Recall that $X/{\approx}$ is defined as the image of $e_{\approx}$ and that this is a subtype of the powerset $\mathcal{P}_{\mathcal{V}}(X)$ which is a set by Lemma 2.7.6. Since it holds generally that subtypes of sets are sets, this proves the lemma. □

Definition 2.11.10 (Universal map, η)**.** The *universal map* $\eta : X \to X/{\approx}$ is defined to be the corestriction of $e_{\approx}$.

Although, in general, the type $X/{\approx}$ lives in another universe than X (see Remark 2.11.8), we can still prove the following induction principle for subsingleton-valued families into *arbitrary* universes.

Lemma 2.11.11 (Set quotient induction)**.** *For every subsingleton-valued family $P : X/{\approx} \to \mathcal{W}$, with $\mathcal{W}$* any *universe, if $P(\eta(x))$ holds for every $x : X$, then $P(x')$ holds for every $x' : X/{\approx}$.*

Proof. The map η is surjective by Lemma 2.6.6, so that Lemma 2.6.7 yields the desired result. □

Definition 2.11.12 (Respect equivalence relation)**.** A map $f : X \to A$ *respects the equivalence relation* $\approx$ if $x \approx y$ implies $f(x) = f(y)$ for every $x, y : X$.

Observe that respecting an equivalence relation is property rather than data, when the codomain A of the map $f : X \to A$ is a set.

Lemma 2.11.13. *The map $\eta : X \to X/{\approx}$ respects the equivalence relation $\approx$ and the set quotient is* effective, *i.e. for every $x, y : X$, we have $x \approx y$ if and only if $\eta(x) = \eta(y)$.*

Proof. By definition of the image and function extensionality, we have for every $x, y : X$ that $\eta(x) = \eta(y)$ holds if and only if

$$\forall_{z:X}(x \approx z \iff y \approx z) \tag{$*$}$$

holds. If ($*$) holds, then so does $x \approx y$ by reflexivity and symmetry of the equivalence relation. Conversely, if $x \approx y$ and $z : X$ is such that $x \approx z$, then $y \approx z$ by symmetry and transitivity; and similarly if $z : X$ is such that $y \approx z$. Hence, ($*$) holds if and only if $x \approx y$ holds. Thus, $\eta(x) = \eta(y)$ if and only if $x \approx y$, as desired. □

The universal property of the set quotient states that the map $\eta : X \to X/{\approx}$ is the universal function to a set preserving the equivalence relation. We can prove it using only Lemma 2.11.11 and Lemma 2.11.13, without needing to inspect the definition of the quotient.

Theorem 2.11.14 (Universal property of the set quotient)**.** *For every* set $A : \mathcal{W}$ *in* any *universe* $\mathcal{W}$ *and function* $f : X \to A$ *respecting the equivalence relation, there is a unique function* $\bar{f} : X/\!\approx\, \to A$ *such that the diagram*

$$\begin{tikzcd} X \arrow[rr, "\eta"] \arrow[dr, "f"'] & & X/\!\approx \arrow[dl, dashed, "\bar{f}"] \\ & A & \end{tikzcd}$$

commutes.

Proof. Let $A : \mathcal{W}$ be a set and $f : X \to A$ respect the equivalence relation. The following auxiliary type family over $X/\!\approx$ will be at the heart of our proof:

$$B(x') :\equiv \Sigma_{a:A} \exists_{x:X}((\eta(x) = x') \times (f(x) = a)).$$

Claim. The type $B(x')$ is a subsingleton for every $x' : X/\!\approx$.

Proof of claim. By function extensionality, the type expressing that $B(x')$ is a subsingleton for every $x' : X/\!\approx$ is itself a subsingleton. So by set quotient induction, it suffices to prove that $B(\eta(x))$ is a subsingleton for every $x : X$. So assume that we have $(a, p), (b, q) : B(\eta(x))$. We only need to show that $a = b$. The elements p and q witness

$$\exists_{x_1:X}((\eta(x_1) = \eta(x)) \times (f(x_1) = a))$$

and

$$\exists_{x_2:X}((\eta(x_2) = \eta(x)) \times (f(x_2) = b)),$$

respectively. By Lemma 2.11.13 and the fact that f respects the equivalence relation, we obtain $f(x) = a$ and $f(x) = b$ and hence the desired $a = b$. □

Next, we define $k : \Pi_{x:X} B(\eta(x))$ by $k(x) = (f(x), |(x, \mathrm{refl}, \mathrm{refl})|)$. By set quotient induction and the claim, the function k induces a dependent map $\bar{k} : \Pi_{(x':X/\approx)} B(x')$. We then define the (nondependent) function $\bar{f} : X/\!\approx\, \to A$ as $\mathrm{pr}_1 \circ \bar{k}$. We proceed by showing that $\bar{f} \circ \eta = f$. By function extensionality, it suffices to prove that we have $\bar{f}(\eta(x)) = f(x)$ for every $x : X$. But notice that:

$$\begin{aligned} \bar{f}(\eta(x)) &\equiv \mathrm{pr}_1(\bar{k}(\eta(x))) \\ &= \mathrm{pr}_1(k(x)) && \text{(since } \bar{k}(\eta(x)) = k(x) \text{ because of the claim)} \\ &\equiv f(x). \end{aligned}$$

Finally, we wish to show that $\bar{f}$ is the unique such function, so suppose that $g : X/\!\approx\, \to A$ is another function such that $g \circ \eta = f$. By function extensionality, it suffices to prove that $g(x') = \bar{f}(x')$ for every $x' : X/\!\approx$, which is a subsingleton as A is a set. Hence, set quotient induction tells us that it is enough to show that $g(\eta(x)) = \bar{f}(\eta(x))$ for every $x : X$, but this holds as both sides of the equation are equal to $f(x)$. □

Remark 2.11.15 (cf. Section 3.21 of [Esc19b]). In univalent foundations, some attention is needed in phrasing unique existence, so we pause to discuss the phrasing of Theorem 2.11.14 here. Typically, if we wish to express unique existence of an element $x : X$ satisfying $P(x)$ for some type family $P : \mathcal{U} \to \mathcal{V}$, then we should phrase it as is-singleton$(\Sigma_{x:X} P(x))$. That is, we require that there is a unique *pair* $(x, p) : \Sigma_{x:X} P(x)$. However, if P is subsingleton-valued, then it is equivalent to the traditional formulation of unique existence: i.e. that there is an $x : X$ with $P(x)$ such that every $y : X$ with $P(y)$ is equal to x. This happens to be the situation in Theorem 2.11.14, because of function extensionality and the fact that A is a set.

We stress that although the set quotient increases universe levels, see Remark 2.11.8, it does satisfy the appropriate universal property, so that resizing is not needed.

Having small set quotients is closely related to propositional resizing, as we show now.

Proposition 2.11.16. *Suppose that $\|-\|$ does not increase universe levels, i.e. in the notation of Remark 2.11.1, the function F is the identity.*

(i) If Ω-Resizing$_{\mathcal{V},\mathcal{U}}$ holds for universes $\mathcal{U}$ and $\mathcal{V}$, then the set quotient $X/\approx$ is $\mathcal{U}$-small for any type $X : \mathcal{U}$ and any $\mathcal{V}$-valued equivalence relation.

(ii) Conversely, if the set quotient $\mathbf{2}/\approx$ is $\mathcal{U}$-small for every $\mathcal{V}$-valued equivalence relation on $\mathbf{2}$, then Propositional-Resizing$_{\mathcal{V},\mathcal{U}}$ *holds.*

Proof. (i): If we have Ω-Resizing$_{\mathcal{V},\mathcal{U}}$, then $\Omega_{\mathcal{V}}$ is $\mathcal{U}$-small, so that $X/\approx \;\equiv \mathrm{im}(e_{\approx})$ is $\mathcal{U}$-small too when $X : \mathcal{U}$ and $\approx$ is $\mathcal{V}$-valued. (ii): Let $P : \mathcal{V}$ be any proposition and consider the $\mathcal{V}$-valued equivalence relation $x \approx_P y :\equiv (x = y) \vee P$ on $\mathbf{2}$. Notice that

$$(\mathbf{2}/\approx_P) \text{ is a subsingleton} \iff P \text{ holds},$$

so if $\mathbf{2}/\approx_P$ is $\mathcal{U}$-small, then so is the type is-subsingleton$(\mathbf{2}/\approx_P)$ and therefore P. $\square$

2.11.3 Propositional truncations from set quotients

Conversely, the propositional truncation arises as a particular set quotient, namely by identifying all elements of a type. However, in order to get an exact match in terms of back-and-forth constructions, we must pay some attention to the universes involved as in Remark 2.11.20 below.

NB. We do not assume the availability of propositional truncations in this section.

Definition 2.11.17 (Existence of set quotients)**.** We say that *set quotients exist* if for every type X and equivalence relation $\approx$ on X, we have a set $X/\approx$ with a universal map $\eta : X \to X/\approx$ that respects the equivalence relation such that the universal property set out in Theorem 2.11.14 is satisfied.

Theorem 2.11.18. *Any set quotient satisfies the induction principle of Lemma 2.11.11, i.e. it is implied by the universal property of the set quotient.*

Proof. Suppose that $P : X/{\approx} \to \mathcal{W}$ is a proposition-valued type-family over the set quotient $X/{\approx}$ and that we have $\rho : \Pi_{x:X} P(\eta(x))$. We write $S :\equiv \Sigma_{x':X/{\approx}} P(x')$ and define the map $f : X \to S$ by $f(x) :\equiv (\eta(x), \rho(x))$. Note that f respects the equivalence relation since η does and P is proposition-valued. Moreover, S is a set, because subtypes of sets are sets and the quotient $X/{\approx}$ is a set by assumption. Hence, by the universal property, f induces a map $\bar{f} : X/{\approx} \to S$ such that $\bar{f} \circ \eta = f$. We claim that $\bar{f}$ is a section of $\mathrm{pr}_1 : S \to X/{\approx}$. Note that this would finish the proof, because if we have $e : \Pi_{x':X/{\approx}} \mathrm{pr}_1\big(\bar{f}(x')\big) = x'$, then we obtain $P(x')$ for every x' by transporting $\mathrm{pr}_2\big(\bar{f}(x')\big)$ along $e(x')$. But $\bar{f}$ must be a section of pr_1, because we can take both $\mathrm{pr}_1 \circ \bar{f}$ and id for the dashed map in the commutative diagram

$$\begin{tikzcd}
X \arrow[rr, "\eta"] \arrow[dr, "\eta"'] & & X/{\approx} \arrow[dl, dashed] \\
& X/{\approx} &
\end{tikzcd}$$

since $\mathrm{pr}_1 \circ \bar{f} \circ \eta = \mathrm{pr}_1 \circ f = \eta$, so $\mathrm{pr}_1 \circ \bar{f}$ and id must be equal by the universal property of the set quotient. □

Theorem 2.11.19. *If set quotients exist, then every type has a propositional truncation.*

Proof. Let $X : \mathcal{U}$ be any type and consider the $\mathcal{U}_0$-valued equivalence relation that identifies all elements: $x \approx_1 y :\equiv \mathbf{1}_{\mathcal{U}_0}$. To see that $X/{\approx_1}$ is a subsingleton, note that by set quotient induction it suffices to prove $\eta(x) = \eta(y)$ for every $x, y : X$. But $x \approx_1 y$ for every $x, y : X$, and η respects the equivalence relation, so this is indeed the case. Now if $P : \mathcal{V}$ is any subsingleton and $f : X \to P$ is any map, then f respects the equivalence relation $\approx_1$ on X, simply because P is a subsingleton. Thus, by the universal property of the quotient, we obtain the desired map $\bar{f} : X/{\approx_1} \to P$ and hence, $X/{\approx_1}$ has the universal property of the propositional truncation. □

Remark 2.11.20. Because the set quotients constructed using the propositional truncation live in higher universes, we embark on a careful comparison of universes. Suppose that propositional truncations of types $X : \mathcal{U}$ exist and that $\|X\| : F(\mathcal{U})$. Then by Remark 2.11.8, the set quotient $X/{\approx_1}$ in the proof above lives in the type universe $(\mathcal{U}_1 \sqcup \mathcal{U}) \sqcup F(\mathcal{U}_1 \sqcup \mathcal{U})$.
In particular, if F is the identity and the propositional truncation of $X : \mathcal{U}$ lives in $\mathcal{U}$, then the quotient $X/{\approx_1}$ lives in $\mathcal{U}_1 \sqcup \mathcal{U}$, which simplifies to $\mathcal{U}$ whenever $\mathcal{U}$ is at least $\mathcal{U}_1$. In other words, the universes of $\|X\|$ and $X/{\approx_1}$ match up for types X in every universe, *except* the first universe $\mathcal{U}_0$.
If we always wish to have $X/{\approx_1}$ in the same universe as $\|X\|$, then we can achieve this by assuming $F(\mathcal{V}) :\equiv \mathcal{U}_1 \sqcup \mathcal{V}$, which says that the propositional truncations stay in the same universe, *except* when the type is in the first universe $\mathcal{U}_0$ in which case the truncation will be in the second universe $\mathcal{U}_1$.

Theorem 2.11.21. *All set quotients are effective, i.e.* $\eta(x) = \eta(y)$ *implies* $x \approx y$.

Proof. If we have set quotients, then we have propositional truncations by Theorem 2.11.19 which we can use to construct effective set quotients following Section 2.11.2. But any two set quotients of a type by an equivalence relation must be equivalent, so the original set quotients are effective too. □

2.11.4 Set replacement

In this section, we return to our running assumption that universes are closed under propositional truncations, i.e. the metafunction F above is assumed to be the identity. We study the equivalence of a set replacement principle and the smallness of set quotients using our construction of Section 2.11.2. These principles will find application in Section 6.3.2, but are quite relevant to us in any case, as smallness of types is a central theme in this thesis.

Definition 2.11.22 (Set replacement)**.** The *set replacement* principle asserts that the image of a map $f : X \to Y$ is $(\mathcal{U} \sqcup \mathcal{V})$-small if X is a $\mathcal{U}$-small type and Y is a locally $\mathcal{V}$-small set.

In particular, if $\mathcal{U}$ and $\mathcal{V}$ are the same, then the image is $\mathcal{U}$-small. The name "set replacement" is inspired by [BBC+22, Section 2.19], but the principle presented here differs from the one in [BBC+22] in two ways: In [BBC+22], replacement is not restricted to maps into sets, and the universe parameters $\mathcal{U}$ and $\mathcal{V}$ are taken to be the same. Rijke [Rij17] shows that the replacement of [BBC+22] is provable in the presence of a univalent universe closed under pushouts.

We are going to show that set replacement is logically equivalent to having small set quotients, where the latter means that the quotient of a type $X : \mathcal{U}$ by a $\mathcal{V}$-valued equivalence relation lives in $\mathcal{U} \sqcup \mathcal{V}$.

Definition 2.11.23 (Existence of small set quotients)**.** We say that *small set quotients exist* if set quotients exists in the sense of Definition 2.11.17, and moreover, the quotient $X/\approx$ of a type $X : \mathcal{U}$ by a $\mathcal{V}$-valued equivalence relation lives in $\mathcal{U} \sqcup \mathcal{V}$.

Note that we would get small set quotients if we added set quotients as a primitive higher inductive type. Also, if one assumes Ω-Resizing$_{\mathcal{V}}$, then the construction of set quotients in Section 2.11.2 yields a quotient $X/\approx$ in $\mathcal{U} \sqcup \mathcal{V}$ when $X : \mathcal{U}$ and $\approx$ is a $\mathcal{V}$-valued equivalence relation on X.

Theorem 2.11.24. *Set replacement is logically equivalent to the existence of small set quotients.*

Proof. Suppose set replacement is true and that a type $X : \mathcal{U}$ and a $\mathcal{V}$-valued equivalence relation $\approx$ are given. Using the construction laid out in Section 2.11.2, we construct a set quotient $X/\approx$ in $\mathcal{U} \sqcup \mathcal{V}^+$ as the image of a map $X \to (X \to \Omega_{\mathcal{V}})$. But by propositional extensionality $\Omega_{\mathcal{V}}$ is locally $\mathcal{V}$-small and by function extensionality so is $X \to \Omega_{\mathcal{V}}$. Hence, $X/\approx$ is $(\mathcal{U} \sqcup \mathcal{V})$-small by set replacement, so $X/\approx$ is equivalent to a type $Y : \mathcal{U} \sqcup \mathcal{V}$. It is then straightforward to show that Y satisfies the properties of the set quotient as well, finishing the proof of one implication.

Conversely, let $f : X \to Y$ be a map from a $\mathcal{U}$-small type to a locally $\mathcal{V}$-small set. Since X is $\mathcal{U}$-small, we have $X' : \mathcal{U}$ such that $X' \simeq X$. And because Y is locally $\mathcal{V}$-small, we have a $\mathcal{V}$-valued binary relation $=_{\mathcal{V}}$ on Y such that $(y =_{\mathcal{V}} y') \simeq (y = y')$ for every $y, y' : Y$. We now define the $\mathcal{V}$-valued equivalence relation $\approx$ on X' by $(x \approx x') :\equiv (f'(x) =_{\mathcal{V}} f'(x'))$, where f' is the composite $X' \simeq X \xrightarrow{f} Y$. By assumption, the quotient $X'/\approx$ lives in $\mathcal{U} \sqcup \mathcal{V}$. But it is straightforward to work out that $\mathrm{im}(f)$ is equivalent to this quotient. Hence, $\mathrm{im}(f)$ is $(\mathcal{U} \sqcup \mathcal{V})$-small, as desired. ☐

The left-to-right implication of the theorem above is similar to [Rij17, Corollary 5.1], but our theorem generalises the universe parameters and restricts to maps into sets. The latter is the reason why the converse also holds.

2.12 Indexed W-types

This final section discusses a general encoding of inductive types known as W-types, which are due to Martin-Löf [Mar84]. The purpose of this encoding is that it allows us to prove results about general inductive types by proving them for W-types. This is exactly what we do in this section. Specifically, we present a criterion for having decidable equality for a further generalisation of W-types known as indexed W-types [GH04; AAG04; AGH+15; Sat15], which will find application in Section 5.2. The further generalisation allows for inductive types with many-sorted constructors as we explain below. But we start by defining and illustrating (nonindexed) W-types.

2.12.1 Basic definitions and examples

Definition 2.12.1 (W-type, $\mathsf{W}_{A,B}$, sup)**.** The W-*type* $\mathsf{W}_{A,B}$ specified by a type $A : \mathcal{U}$ and type family $B : A \to \mathcal{V}$ is the inductive type with a single constructor

$$\mathrm{sup} : \Pi_{a:A}\big((B(a) \to \mathsf{W}_{A,B}) \to \mathsf{W}_{A,B}\big).$$

We postulate that $\mathsf{W}_{A,B}$ lives in the universe $\mathcal{U} \sqcup \mathcal{V}$.

Remark 2.12.2 (The induction principle of a W-type). Spelling out the induction principle of $\mathsf{W}_{A,B}$, it reads: for every $Y : \mathsf{W}_{A,B} \to \mathcal{T}$, then to prove $Y(w)$ for every $w : \mathsf{W}_{A,B}$, it suffices to prove that for any $a : A$ and $f : B(a) \to \mathsf{W}_{A,B}$ satisfying $Y(f(b))$ for every $b : B(a)$ (the "induction hypothesis"), we have $Y(\mathrm{sup}(a, f))$.

The elements of a W-type $\mathsf{W}_{A,B}$ can be thought of as some kind of well-founded trees, hence the name W-type, but we prefer another viewpoint. We think of the type A in the definition above as a type of labels for constructors, while $B(a)$ encodes the arity of the constructor labelled by a.[1] It is instructive to see how W-types can encode the type of natural numbers.

[1]The name sup does not make much sense from this point of view, but it is the traditional name in the existing literature.

Example 2.12.3 (The type of natural numbers as a W-type)**.** Following the above description, we define $A :\equiv \mathbf{2}$, since $\mathbb{N}$ has two constructors. Furthermore, we put $B(0) = \mathbf{0}$, since the $\texttt{zero}$ constructor takes no arguments, while $B(1) = \mathbf{1}$, because $\texttt{succ}$ takes one recursive argument.
We recursively define functions back and forth between the types as follows:

$$\begin{aligned} \phi : \mathbb{N} &\to \mathsf{W}_{A,B} & \psi : \mathsf{W}_{A,B} &\to \mathbb{N} \\ \texttt{zero} &\mapsto \sup(0, \text{unique-from-}\mathbf{0}), & \sup(0, f) &\mapsto \texttt{zero}, \\ \texttt{succ}(n) &\mapsto \sup(1, \lambda\star . \phi(n)), & \sup(1, f) &\mapsto \texttt{succ}(\psi(f(\star))), \end{aligned}$$

where unique-from-$\mathbf{0}$ is the unique map from $\mathbf{0}$ to $\mathsf{W}_{A,B}$.
Using the induction principles of $\mathbb{N}$ and $\mathsf{W}_{A,B}$ it is then straightforward to prove that ϕ and ψ are inverses. Thus, the types $\mathbb{N}$ and $\mathsf{W}_{A,B}$ are equivalent.

Another example that will come in useful later when studying the programming language PCF is the encoding of PCF types.

Example 2.12.4 (The PCF types as a W-type)**.** The PCF types are inductively generated: ι is a PCF type, known as the *base type* and if σ and τ are PCF types, then we have another PCF type, called the *(PCF) function type* and denoted by $\sigma \Rightarrow \tau$.
We can encode this inductive type as a W-type as follows. Take $A :\equiv \mathbf{2}$ since we have two constructors and put $B(0) = \mathbf{0}$, since the base type needs no arguments, while $B(1) = \mathbf{2}$, because to construct a PCF function type we need to be given two PCF types. The maps back and forth the inductive types are given by

$$\begin{aligned} \phi : \text{PCF-Types} &\to \mathsf{W}_{A,B}, & \psi : \mathsf{W}_{A,B} &\to \text{PCF-Types} \\ \iota &\mapsto \sup(0, \text{unique-from-}\mathbf{0}), & \sup(0, f) &\mapsto \iota, \\ \sigma \Rightarrow \tau &\mapsto \sup(1, g), & \sup(1, f) &\mapsto \psi(f(0)) \Rightarrow \psi(f(1)), \end{aligned}$$

with $g : \mathbf{2} \to \mathsf{W}_{A,B}$ given by $g(0) :\equiv \phi(\sigma)$ and $g(1) :\equiv \phi(\tau)$. Using the induction principles of PCF-Types and $\mathsf{W}_{A,B}$ it is then straightforward to prove that ϕ and ψ are inverses. Thus, the types PCF-Types and $\mathsf{W}_{A,B}$ are equivalent.

We now generalise W-types to indexed W-types which encode inductively defined *families* over some type I.

Definition 2.12.5 (Indexed W-type, $\mathsf{W}_{s,t}$, sup)**.** Let A and I be types and let B be a type family over A. Suppose we have $t : A \to I$ and $s : (\Sigma_{a:A} B(a)) \to I$. The *indexed* W*-type* $\mathsf{W}_{s,t}$ specified by s and t is the inductive type family over I generated by the constructor

$$\sup : \prod_{a:A} \Big(\big(\Pi_{b:B(a)} \mathsf{W}_{s,t}(s(a,b)) \big) \to \mathsf{W}_{s,t}(t(a)) \Big).$$

If we have $I : \mathcal{U}$, $A : \mathcal{V}$ and $B : A \to \mathcal{W}$, then we assume $\mathsf{W}_{s,t}(i) : \mathcal{V} \sqcup \mathcal{W}$ for all $i : I$.

Remark 2.12.6 (The induction principle of an indexed W-type). We spell out the induction principle for indexed W-types. If we have $Y : \Pi_{i:I}\big(\mathsf{W}_{s,t}(i) \to \mathcal{T}\big)$, then to prove

$\Pi_{i:I}\Pi_{w:\mathsf{W}_{s,t}(i)}Y(i,w)$, it suffices to show that for any $a : A$ and $f : \Pi_{b:B(a)}\mathsf{W}_{s,t}(s(a,b))$ satisfying $Y(s(a,b), f(b))$ for every $b : B(a)$ (the *induction hypothesis*), we have a term of type $Y(t(a), \mathrm{sup}(a, f))$.

That this is indeed a generalisation of W-types is witnessed by the following result.

Proposition 2.12.7. *Every* W*-type is equivalent to an indexed* W*-type over* $\mathbf{1}$.

Proof. Given a W-type with parameters A and B, we define the functions $t : A \to \mathbf{1}$ and $s : (\Sigma_{a:A}B(a)) \to \mathbf{1}$ to be the unique maps to $\mathbf{1}$. It is then not hard to see that $\mathsf{W}_{A,B} \simeq \mathsf{W}_{s,t}(\star) \simeq \Sigma_{u:\mathbf{1}}\mathsf{W}_{s,t}(u)$. ☐

As with ordinary W-types, we think of A as a type of (labels of) constructors and of B as encoding the arity of the constructors. But in the indexed case each constructor $a : A$ has a sort given by $t(a) : I$ and the arguments $b : B(a)$ to a constructor $a : A$ also have sorts given by $s(a,b)$.[2] Again, it is illuminating to look at an example.

Example 2.12.8 (A subset of PCF as an indexed W-type)**.** In this example, we define the terms of a very basic typed programming language, using the PCF types of Example 2.12.4. This will be a subset of the PCF programming language studied in Section 5.2.1.
The type family $\mathsf{T} : \text{PCF-Types} \to \mathcal{U}_0$ is inductively defined as:
(i) zero is a term of type ι (i.e. $\mathtt{zero} : \mathsf{T}(\iota)$);
(ii) succ is a term of type $\iota \Rightarrow \iota$;
(iii) for every two PCF types σ and τ, given a term s of type $\sigma \Rightarrow \tau$ and a term t of type σ, we have a term of type τ denoted by juxtaposition st and called s *applied to* t.

Besides application our only other terms are zero and succ, so the fact that our application defined for general types doesn't get us very much, but it helps to illustrate indexed W-types. For the indexed W-type, take I to be the type of PCF types and put $A :\equiv \mathbf{2} + (I \times I)$. Define $B : A \to \mathcal{U}_0$ by

$$\begin{aligned} B(\mathrm{inl}(0)) :\equiv B(\mathrm{inl}(1)) &:\equiv \mathbf{0}, \text{ and} \\ B(\mathrm{inr}(\sigma,\tau)) &:\equiv \mathbf{2}. \end{aligned}$$

Finally, define t by

$$\begin{aligned} t(\mathrm{inl}(0)) &:\equiv \iota, \\ t(\mathrm{inl}(1)) &:\equiv \iota \Rightarrow \iota, \text{ and} \\ t(\mathrm{inr}(\sigma,\tau)) &:\equiv \tau; \end{aligned}$$

and s by

$$\begin{aligned} s(\mathrm{inr}(\sigma,\tau),0) &:\equiv \sigma \Rightarrow \tau, \text{ and} \\ s(\mathrm{inr}(\sigma,\tau),1) &:\equiv \sigma; \end{aligned}$$

[2]The names s and t stand for source and target respectively.

on the other elements s is defined as the unique function from $\mathbf{0}$.
It is then tedious, but straightforward to define, by induction on PCF types, maps between $\mathsf{T}(\sigma)$ and $\mathsf{W}_{s,t}(\sigma)$ for every PCF type σ and prove that they are inverses, establishing that $\mathsf{T}(\sigma)$ and $\mathsf{W}_{s,t}(\sigma)$ are equivalent for every PCF type σ.

2.12.2 Indexed W-types with decidable equality

We wish to isolate some conditions on the parameters of an indexed W-type that are sufficient to conclude that an indexed W-type has decidable equality. We first need a few definitions before we can state the theorem.

Definition 2.12.9 (Π-compactness; [Esc+, `TypeTopology.CompactTypes`])**.** A type X is Π-*compact* when every type family Y over X satisfies: if $Y(x)$ is decidable for every $x : X$, then so is the dependent product $\Pi_{x:X} Y(x)$.

Example 2.12.10. The empty type $\mathbf{0}$ is vacuously Π-compact. The unit type $\mathbf{1}$ is also easily seen to be Π-compact. There are also interesting examples of infinite types that are Π-compact, such as $\mathbb{N}_\infty$, the one-point compactification of the natural numbers [Esc19a].

Lemma 2.12.11 (cf. [Esc+, `TypeTopology.CompactTypes`])**.** *The Π-compact types are closed under binary coproducts.*

Proof. Let X and Y be Π-compact types. Suppose F is a type family over $X + Y$ such that $F(z)$ is decidable for every $z : X + Y$. We must show that $\Pi_{z:X+Y} F(z)$ is decidable. Define the type family F_X over X by $F_X(x) :\equiv F(\mathrm{inl}(x))$ and the type family F_Y over Y by $F_Y(y) :\equiv F(\mathrm{inr}(y))$. By our assumption on F, the types $F_X(x)$ and $F_Y(y)$ are decidable for every $x : X$ and $y : Y$. Hence, since X and Y are assumed to be Π-compact, the dependent products $\Pi_{x:X} F_X(x)$ and $\Pi_{y:Y} F_Y(y)$ are decidable.
Finally, $\Pi_{z:X+Y} F(z)$ is logically equivalent to $(\Pi_{x:X} F_X(x)) \times \big(\Pi_{y:Y} F_Y(y)\big)$. Since the product of two decidable types is again decidable, an application of Lemma 2.7.11 now finishes the proof. ☐

We are now in position to state the general theorem about decidable equality on indexed W-types. We will use this theorem in Section 5.2 to prove that the syntax of the typed programming language PCF has decidable equality.

Theorem 2.12.12. *An indexed W-type $\mathsf{W}_{s,t}$ specified by parameters $s : A \to I$ and $t : (\Sigma_{a:A} B(a)) \to I$ has decidable equality at every $i : I$ if*

(i) I is a set,

(ii) A has decidable equality, and

(iii) $B(a)$ is Π-compact for every $a : A$.

Corollary 2.12.13. *A W-type $\mathsf{W}_{A,B}$ specified by a type A and a type family B over A has decidable equality if A has decidable equality and $B(a)$ is Π-compact for every $a : A$.*

Proof. By Proposition 2.12.7 the type $\mathsf{W}_{A,B}$ is an indexed W-type over $I :\equiv \mathbf{1}$ which is a set. □

The proof of Theorem 2.12.12 is quite technical, so we postpone it until Section 2.12.3. Instead, we next describe how to apply the theorem to prove that the PCF types and the type family from Example 2.12.8 have decidable equality.

Proposition 2.12.14. *The type family from Example 2.12.8 has decidable equality.*

Proof. We apply Theorem 2.12.12 with parameters $t : A \to I$ and $s : (\Sigma_{a:A} B(a)) \to I$ for the indexed W-type defined in Example 2.12.8. By using Example 2.12.10 and Lemma 2.12.11 we see that $B(a)$ is Π-compact for every $a : A$. Further, note that $A \equiv I$ has decidable equality if I does. So it remains to prove that I, the type of PCF types, has decidable equality. But this follows from Corollary 2.12.13, because by Example 2.12.4 we can encode PCF types as a (nonindexed) W-type with parameters $A :\equiv \mathbf{2}$ and $B : A \to \mathcal{U}_0$ given by $B(0) :\equiv \mathbf{0}$ and $B(1) :\equiv \mathbf{2}$, and A has decidable equality while $B(a)$ is Π-compact for every $a : A$ because of Example 2.12.10 and Lemma 2.12.11. □

2.12.3 Proving indexed W-types to have decidable equality

In this section we prove Theorem 2.12.12 by deriving it as a corollary of another result, namely Theorem 2.12.15 below. This result seems to have been first established by Jasper Hugunin, who reported on it in a post on the *Homotopy Type Theory* mailing list [Hug17a]. Our proof of Theorem 2.12.15 is a simplified written-up account of Hugunin's CoQ code [Hug17b, `FiberProperties.v`].

Theorem 2.12.15 (Jasper Hugunin)**.** *An indexed W-type $\mathsf{W}_{s,t}$ specified by $s : A \to I$ and $t : (\Sigma_{a:A} B(a)) \to I$ has decidable equality at every $i : I$ if*

(i) $B(a)$ is Π-compact for every $a : A$, and

(ii) the fibres of t at every $i : I$ all have decidable equality.

Let us see how to obtain Theorem 2.12.12 from Theorem 2.12.15.

Proof of Theorem 2.12.12 (using Theorem 2.12.15). Suppose that A has decidable equality and I is a set. We are to show that the fibre of t over i has decidable equality for every $i : I$. So suppose we have (a, p) and (a', p') in the fibre of t over i. Since A has decidable equality, we can decide whether a and a' are equal or not. If they are not, then certainly $(a, p) \neq (a', p')$. If they are, then we claim that the dependent pairs (a, p) and (a', p') are also equal. If $e : a = a'$ is the supposed equality, then it suffices to show that $\mathsf{transport}^{\lambda x:A.t(x)=i}(e, p) = p'$, but both these elements are identifications in I and I is a set, so they must be equal. □

We now embark on a proof of Theorem 2.12.15. For the remainder of this section, let us fix types A and I, a type family B over A and maps $t : A \to I$ and $s : (\Sigma_{a:A} B(a)) \to I$.

We do not prove the theorem directly. The statement makes it impossible to assume two elements $u, v : \mathsf{W}_{s,t}(i)$ and proceed by induction on *both* u and v. Instead, we will state and prove a more general result that is amenable to a proof by induction. But first, we need additional general lemmas and some definitions.

Lemma 2.12.16. *If X is a set, then the right pair function of any type family Y over X is left-cancellable, in the following sense: if $(x, y) = (x, y')$ as elements of $\Sigma_{a:X} Y(a)$, then $y = y'$.*

Proof. Suppose X is a set, $x : X$ and $y, y' : Y(x)$ with $e : (x, y) = (x, y')$. From e, we obtain $e_1 : x = x$ and $e_2 : \mathrm{transport}^Y(e_1, y) = y'$. Since X is a set, we must have that $e_1 = \mathrm{refl}_x$, so that e_2 yields $y \equiv \mathrm{transport}^Y(\mathrm{refl}_x, y) = y'$, as desired. □

Definition 2.12.17 (Subtrees, sub_i)**.** For each $i : I$, define

$$\mathrm{sub}_i : \mathrm{W}_{s,t}(i) \to \sum_{p:\mathrm{fib}_t(i)} \prod_{b:B(\mathrm{pr}_1(p))} \mathrm{W}_{s,t}(s(\mathrm{pr}_1(p), b))$$

by induction as

$$\mathrm{sub}_{t(a)}(\mathrm{sup}(a, f)) :\equiv \big((a, \mathrm{refl}_{t(a)}), f\big).$$

For notational convenience, we will omit the subscript of sub.

The name sub comes from *sub*trees, thinking of the elements of a W-type as a well-founded trees.

Lemma 2.12.18. *If the fibre of t over i has decidable equality for every $i : I$, then $\mathrm{sup}(a, f) = \mathrm{sup}(a, g)$ implies $f = g$ for every $a : A$ and $f, g : \Pi_{b:B(a)} \mathrm{W}_{s,t}(s(a, b))$.*

Proof. Suppose $\mathrm{sup}(a, f) = \mathrm{sup}(a, g)$. Then

$$\big((a, \mathrm{refl}_{t(a)}), f\big) \equiv \mathrm{sub}(\mathrm{sup}(a, f)) = \mathrm{sub}(\mathrm{sup}(a, g)) \equiv \big((a, \mathrm{refl}_{t(a)}), g\big).$$

As $\mathrm{fib}_t(i)$ is assumed to be decidable, it is a set by Hedberg's theorem (Theorem 2.7.12). Hence $f = g$ by Lemma 2.12.16. □

Definition 2.12.19 (to-fib_i)**.** For every $i : I$, we define $\text{to-fib}_i : \mathrm{W}_{s,t}(i) \to \mathrm{fib}_t(i)$ inductively by

$$\text{to-fib}_{t(a)}(\mathrm{sup}(a, f)) :\equiv (a, \mathrm{refl}_{t(a)}).$$

In future use, we omit the subscript of to-fib.

Lemma 2.12.20. *For $i, j : I$ with a path $p : i = j$ and $w : \mathrm{W}_{s,t}(i)$, we have the following equality:*

$$\text{to-fib}(\mathrm{transport}^{\mathrm{W}_{s,t}}(p, w)) = (\mathrm{pr}_1(\text{to-fib}(w)), \mathrm{pr}_2(\text{to-fib}(w)) \bullet p).$$

Proof. By path induction on p. □

We are now in position to state and prove the lemma from which we will derive Theorem 2.12.15.

Lemma 2.12.21. *Suppose that $B(a)$ is Π-compact for every $a : A$ and that the fibre of t over each $i : I$ has decidable equality. For any $i : I$, $u : \mathsf{W}_{s,t}(i)$, $j : I$, path $p : i = j$ and $v : \mathsf{W}_{s,t}(j)$, the type*

$$\mathsf{transport}^{\mathsf{W}_{s,t}}(p, u) = v$$

is decidable.

Proof. Suppose $i : I$ and $u : \mathsf{W}_{s,t}(i)$. We proceed by induction on u and so we assume that $u \equiv \mathsf{sup}(a, f)$. The induction hypothesis reads:

$$\prod_{b:B(a)} \prod_{j':I} \prod_{p':s(a,b)=j'} \prod_{v':\mathsf{W}_{s,t}(j')} (\mathsf{transport}^{\mathsf{W}_{s,t}}(p', f(b)) = v') \text{ is decidable .} \tag{$*$}$$

Suppose we have $j : I$ with path $p : t(a) = j$ and $v : \mathsf{W}_{s,t}(j)$. By induction, we may assume that $v \equiv \mathsf{sup}(a', f')$. We are tasked to show that

$$\mathsf{transport}^{\mathsf{W}_{s,t}}(p, \mathsf{sup}(a, f)) = \mathsf{sup}(a', f') \tag{$\dagger$}$$

is decidable, where $p : t(a) = t(a')$.
By assumption the fibre of t over $t(a')$ has decidable equality. Hence, we can decide if $(a', \mathsf{refl}_{t(a')})$ and (a, p) are equal or not. Suppose first that the pairs are not equal. We claim that in this case $\neg(\dagger)$. For suppose we had $e : (\dagger)$, then

$$\mathsf{ap}_{\mathsf{to\text{-}fib}}(e) : \mathsf{to\text{-}fib}(\mathsf{transport}^{\mathsf{W}_{s,t}}(p, \mathsf{sup}(a, f))) = \mathsf{to\text{-}fib}(\mathsf{sup}(a', f')).$$

By definition, the right hand side is $(a', \mathsf{refl}_{t(a')})$. By Lemma 2.12.20, the left hand side is equal to $(a, \mathsf{refl}_{t(a)} \bullet p)$ which is in turn equal to (a, p), contradicting our assumption that $(a', \mathsf{refl}_{t(a')})$ and (a, p) were not equal.
Now suppose that $(a', \mathsf{refl}_{t(a')}) = (a, p)$. From this, we obtain paths $e_1 : a' = a$ and $e_2 : \mathsf{transport}^{\lambda x:A.t(x)=t(a')}(e_1, \mathsf{refl}_{t(a')}) = p$. By path induction, we may assume $e_1 \equiv \mathsf{refl}_{a'}$, so that from e_2 we obtain an identification

$$\rho : \mathsf{refl}_{t(a')} = p.$$

Using this identification, we see that the left hand side of ($\dagger$) is equal to $\mathsf{sup}(a', f)$, so we are left to show that

$$\mathsf{sup}(a', f) = \mathsf{sup}(a', f')$$

is decidable. By induction hypothesis ($*$) and the fact that $a \equiv a'$, the type $f(b) = f'(b)$ is decidable for every $b : B(a')$. Since $B(a')$ is Π-compact, this implies that the type $\left(\Pi_{b:B(a')} f(b) = f'(b)\right)$ is decidable. Suppose first that $\Pi_{b:B(a')} f(b) = f'(b)$. Function extensionality then yields $f = f'$, so that $\mathsf{sup}(a', f) = \mathsf{sup}(a', f')$. On the other hand, suppose $\neg\left(\Pi_{b:B(a')} f(b) = f'(b)\right)$. We claim that the elements $\mathsf{sup}(a', f)$ and $\mathsf{sup}(a', f')$ cannot be equal. For if they were equal, then Lemma 2.12.18 would yield $f = f'$, contradicting our assumption that $\neg\left(\Pi_{b:B(a)} f(b) = f'(b)\right)$, which finishes the proof. □

The proof of Theorem 2.12.15 now follows readily.

Proof of Theorem 2.12.15. Let $i : I$ and $u, v : \mathsf{W}_{s,t}(i)$. Taking $j :\equiv i$ and $p :\equiv \mathsf{refl}_i$ in Lemma 2.12.21, we see that $u = v$ is decidable, as desired. $\square$

2.13 Notes

Our discussion of type universes in Section 2.1 closely follows that of [Esc21, Section 2.1]. Our treatment of univalent foundations in Sections 2.2–2.8 is our own, but based on the expositions in [Uni13] and [Esc19b]. The notion of a (locally) $\mathcal{U}$-small type appears in [Rij17], but the lemmas in Section 2.9 involving retracts are our original results and were included in our paper [dJE21b] and its extended version [dJE22a].

Section 2.11.1 closely follows the exposition in [Esc19b, Section 3.34.1], while Section 2.11.2 is based on [Esc19b, Section 3.37]. Both Sections 2.11.3 and 2.11.4 are original contributions. Section 2.11 as a whole was included in our work [dJE21b; dJE22a].

Finally, W-types, studied in Section 2.12, were introduced by Per Martin-Löf [Mar84] and the main theorem presented in that section is, as mentioned before, due to Jasper Hugunin [Hug17b; Hug17a]. This result was also included in our paper [dJon21b] with an application to semidecidability questions pertaining to the Scott model of PCF (as discussed in Section 5.2 of this thesis).

Chapter 3

Basic domain theory

Domain theory [AJ94] is a well-established subject in mathematics and theoretical computer science with applications to programming language semantics [Sco72; Sco93; Plo77], higher-type computability [LN15], topology, and more [GHK+03]. In this chapter we introduce basic domain theory within the context of constructive and predicative univalent foundations. Specifically, we discuss

Section 3.2: (pointed) dcpos: the basic objects of domain theory,
Section 3.3: Scott continuous maps: morphism between dcpos,
Section 3.4: the lifting of a set and of a dcpo to get the free pointed dcpo,
Section 3.5: products and exponentials of dcpos, and
Section 3.6: bilimits of dcpos.

The basic theory will find application in the semantics of programming languages, as laid out in Chapter 5. We offer the following overture in preparation of our development, especially if the reader is familiar with domain theory in a classical, set-theoretic setting.

3.1 Introduction

The basic object of study in domain theory is that of a *directed complete poset* (dcpo). In (impredicative) set-theoretic foundations, a dcpo can be defined to be a poset that has least upper bounds of all directed subsets. A naive translation of this to our foundation would be to proceed as follows. Define a poset in a universe $\mathcal{U}$ to be a type $P : \mathcal{U}$ with a reflexive, transitive and antisymmetric relation $- \sqsubseteq - : P \times P \to \mathcal{U}$. Since we wish to consider posets and not categories we require that the values $p \sqsubseteq q$ of the order relation are *subsingletons*. Then we could say that the poset $(P, \sqsubseteq)$ is *directed complete* if every directed family $I \to P$ with indexing type $I : \mathcal{U}$ has a least upper bound (supremum). The problem with this definition is that there are no interesting examples in our constructive and predicative setting. For instance, assume that the poset $\mathbf{2}$ with two elements $0 \sqsubseteq 1$ is directed complete, and consider a proposition $A : \mathcal{U}$ and the directed family $A + \mathbf{1} \to \mathbf{2}$ that maps the left component to 0 and the right component to 1. By case analysis on its hypothetical supremum (Definition 3.2.6), we conclude that the negation of A is decidable. This amounts to weak excluded middle (Definition 2.7.19) which is constructively unacceptable.

To try to get an example, we may move to the poset $\Omega_{\mathcal{U}_0}$ of propositions in the universe $\mathcal{U}_0$, ordered by implication. This poset does have all suprema of families $I \to \Omega_{\mathcal{U}_0}$ indexed by types I in the *first universe* $\mathcal{U}_0$, given by existential quantification. But if we consider a directed family $I \to \Omega_{\mathcal{U}_0}$ with I in the *same universe* as $\Omega_{\mathcal{U}_0}$ lives, namely the *second universe* $\mathcal{U}_1$, existential quantification gives a proposition in the *second universe* $\mathcal{U}_1$ and so doesn't give its supremum. In this example, we get a poset such that

(i) the carrier lives in the universe $\mathcal{U}_1$,
(ii) the order has truth values in the universe $\mathcal{U}_0$, and
(iii) suprema of directed families indexed by types in $\mathcal{U}_0$ exist.

Regarding a poset as a category in the usual way, we have a large, but locally small, category with small filtered colimits (directed suprema). This is typical of all the concrete examples that we will consider, such as the dcpos in the Scott model of PCF (Section 5.2) and Scott's D_∞ model of the untyped λ-calculus (Section 5.1). We may say that the predicativity restriction increases the universe usage by one. However, for the sake of generality, we formulate our definition of dcpo with the following universe conventions:

(i) the carrier lives in a universe $\mathcal{U}$,
(ii) the order has truth values in a universe $\mathcal{T}$, and
(iii) suprema of directed families indexed by types in a universe $\mathcal{V}$ exist.

So our notion of dcpo has three universe parameters $\mathcal{U}, \mathcal{V}$ and $\mathcal{T}$. We will say that the dcpo is *locally small* when $\mathcal{T}$ is not necessarily the same as $\mathcal{V}$, but the order has $\mathcal{V}$-small (recall Definition 2.9.1) truth values. Most of the time we mention $\mathcal{V}$ explicitly and leave $\mathcal{U}$ and $\mathcal{T}$ to be understood from the context.

3.2 Directed complete posets

We introduce the basic object of domain theory: a directed complete poset. We carefully explain our use of the propositional truncation in our definitions and, as mentioned above, the type universes involved.

Definition 3.2.1 (Preorder, reflexivity, transitivity)**.** A *preorder* $(P, \sqsubseteq)$ is a type $P : \mathcal{U}$ together with a proposition-valued binary relation $\sqsubseteq : P \to P \to \Omega_{\mathcal{T}}$ satisfying

(i) *reflexivity*: for every $p : P$, we have $p \sqsubseteq p$, and
(ii) *transitivity*: for every $p, q, r : P$, if $p \sqsubseteq q$ and $q \sqsubseteq r$, then $p \sqsubseteq r$.

Definition 3.2.2 (Poset, antisymmetry)**.** A *poset* is a preorder $(P, \sqsubseteq)$ that is *antisymmetric*: if $p \sqsubseteq q$ and $q \sqsubseteq p$, then $p = q$ for every $p, q : P$.

Lemma 3.2.3. *If* $(P, \sqsubseteq)$ *is a poset, then* P *is a set.*

Proof. For every $p, q : P$, the composite

$$(p = q) \xrightarrow{\text{by reflexivity}} (p \sqsubseteq q) \times (q \sqsubseteq p) \xrightarrow{\text{by antisymmetry}} (p = q)$$

is constant, since $(p \sqsubseteq q) \times (q \sqsubseteq p)$ is a proposition. Hence, by Lemma 2.3.13 it follows that P must be a set. □

From now on, we will simply write "let P be a poset" leaving the partial order $\sqsubseteq$ implicit. We will often use the symbol $\sqsubseteq$ for partial orders on different carriers when it is clear from the context which one it refers to.

Definition 3.2.4 ((Semi)directed family)**.** A family $\alpha : I \to P$ of elements of a poset P is *semidirected* if whenever we have $i, j : I$, there exists some $k : I$ such that $\alpha_i \sqsubseteq \alpha_k$ and $\alpha_j \sqsubseteq \alpha_k$. We frequently use the shorthand $\alpha_i, \alpha_j \sqsubseteq \alpha_k$ to denote the latter requirement. Such a family is *directed* if it is semidirected and its domain I is inhabited.

The name "semidirected" matches Taylor's terminology [Tay99, Definition 3.4.1].

Remark 3.2.5. Note our use of the propositional truncation in defining when a family is *directed*. To make this explicit, we write out the definition in type-theoretic syntax: a family $\alpha : I \to P$ is directed if

(i) we have an element of $\|I\|$, and

(ii) $\Pi_{i,j:I} \left\| \Sigma_{k:I} (\alpha_i \sqsubseteq \alpha_k) \times \left(\alpha_j \sqsubseteq \alpha_k\right) \right\|$.

The use of the propositional truncation ensures Item (i) and Item (ii) are propositions and hence that being (semi)directed is a property of a family. Item (ii) without truncating would be asking us to assign a chosen $k : I$ for every $i, j : I$ instead.

Following Scott [Sco70], we sometimes think of the elements of P as pieces of information and $p \sqsubseteq q$ as expressing that q contains more information or refines p. With this viewpoint, a directed family is a collection of pieces of information that are consistent in the sense that any two pieces of information can be refined to a third one that is a member of the collection. In the next definition we ask for such families to have least upper bounds, which is like saying that such consistent collections of information can be patched together to a piece of information that refines everything in the family.

Definition 3.2.6 ((Least) upper bound, supremum)**.** An element x of a poset P is an *upper bound* of a family $\alpha : I \to P$ if $\alpha_i \sqsubseteq x$ for every $i : I$. It is a *least upper bound* of α if it is an upper bound, and whenever $y : P$ is an upper bound of α, then $x \sqsubseteq y$. By antisymmetry, a least upper bound is unique if it exists, so in this case we will speak of *the* least upper bound of α, or sometimes the *supremum* of α.

Definition 3.2.7 ($\mathcal{V}$-directed complete poset, $\mathcal{V}$-dcpo, $\bigsqcup \alpha$, $\bigsqcup_{i:I} \alpha_i$)**.** For a universe $\mathcal{V}$, a *$\mathcal{V}$-directed complete poset* (or *$\mathcal{V}$-dcpo*, for short) is a poset D such that every directed family $\alpha : I \to D$ with $I : \mathcal{V}$ has a supremum in D that we denote by $\bigsqcup \alpha$ or $\bigsqcup_{i:I} \alpha_i$.

Remark 3.2.8. Explicitly, we ask for an element of the type

$$\Pi_{I:\mathcal{V}}\Pi_{\alpha:I\to D}(\text{is-directed}\,\alpha \to \Sigma_{x:D}(x \text{ is-sup-of } \alpha)),$$

where $(x \text{ is-sup-of } \alpha)$ is the type expressing that x is the supremum of α. Even though we used Σ and not $\exists$ in this expression, this type is still a proposition: By Theorem 2.3.17, it suffices to prove that the type $\Sigma_{x:D}(x \text{ is-sup-of } \alpha)$ is a proposition. So suppose that we have $x, y : D$ with $p : x \text{ is-sup-of } \alpha$ and $q : y \text{ is-sup-of } \alpha$. Being the supremum of a family is a property because the partial order is proposition-valued. Hence, by Lemma 2.4.8, to prove that $(x, p) = (y, q)$, it suffices to prove that $x = y$. But this follows from antisymmetry and the fact that x and y are both suprema of α.

We will sometimes leave the universe $\mathcal{V}$ implicit, and simply speak of a dcpo. On other occasions, we need to carefully keep track of universe levels. To this end, we make the following definition.

Definition 3.2.9 ($\mathcal{V}$-DCPO$_{\mathcal{U},\mathcal{T}}$)**.** Let $\mathcal{V}, \mathcal{U}$ and $\mathcal{T}$ be universes. We write $\mathcal{V}$-DCPO$_{\mathcal{U},\mathcal{T}}$ for the type of $\mathcal{V}$-dcpos with carrier in $\mathcal{U}$ and order taking values in $\mathcal{T}$.

Remark 3.2.10. In particular, it is very important to keep track of the universe parameters of the lifting (Section 3.4) and of exponentials (Section 3.5) in order to ensure that it is possible to construct the Scott model of PCF and Scott's D_∞ in our predicative setting, as we do in Chapter 5.

In many examples and applications, we require our dcpos to have a least element.

Definition 3.2.11 (Pointed dcpo)**.** A dcpo D is *pointed* if it has a least element which we will denote by $\bot_D$, or simply $\bot$.

Definition 3.2.12 (Local smallness)**.** A $\mathcal{V}$-dcpo D is *locally small* if $x \sqsubseteq y$ is $\mathcal{V}$-small (recall Definition 2.9.1) for every $x, y : D$.

Lemma 3.2.13. *A $\mathcal{V}$-dcpo D is locally small if and only if we have $\sqsubseteq_{\mathcal{V}} : D \to D \to \mathcal{V}$ such that $x \sqsubseteq y$ holds precisely when $x \sqsubseteq_{\mathcal{V}} y$ does.*

Proof. The $\mathcal{V}$-dcpo D is locally small exactly when we have an element of

$$\Pi_{x,y:D}\Sigma_{T:\mathcal{V}}(T \simeq x \sqsubseteq y).$$

But this type is equivalent to

$$\Sigma_{R:D\to D\to\mathcal{V}}\Pi_{x,y:D}(R(x,y) \simeq x \sqsubseteq y)$$

by Lemma 2.6.2. $\square$

Nearly all examples of $\mathcal{V}$-dcpos in this thesis will be locally small. We now introduce two fundamental examples of dcpos: the type of subsingletons and powersets.

Example 3.2.14 (The type of subsingletons as a pointed dcpo)**.** For any type universe $\mathcal{V}$, the type $\Omega_{\mathcal{V}}$ of subsingletons in $\mathcal{V}$ is a poset if we order the propositions by implication. Note that antisymmetry holds precisely because of propositional extensionality (Definition 2.3.20). Moreover, $\Omega_{\mathcal{V}}$ has a least element, namely $\mathbf{0}_{\mathcal{V}}$, the empty type in $\mathcal{V}$. We also claim that $\Omega_{\mathcal{V}}$ has suprema for all (not necessarily directed) families $\alpha : I \to \Omega_{\mathcal{V}}$ with $I : \mathcal{V}$. Given such a family α, its least upper bound is given by $\exists_{i:I}\, \alpha_i$. It is clear that this is indeed an upper bound for α. And if P is a subsingleton such that $\alpha_i \sqsubseteq P$ for every $i : I$, then to show that $(\exists_{i:I}\, \alpha_i) \to P$ it suffices to construct to construct a map $(\Sigma_{i:I}\, \alpha_i) \to P$ as P is a subsingleton. But this is easy because we assumed that $\alpha_i \sqsubseteq P$ for every $i : I$. Finally, paying attention to the universe levels we observe that $\Omega_{\mathcal{V}} : \mathcal{V}\text{-DCPO}_{\mathcal{V}^+,\mathcal{V}}$.

Example 3.2.15 (Powersets as pointed dcpos)**.** Recalling our treatment of subset and powersets from Section 2.7.1, we show that powersets give examples of pointed dcpos. Specifically, for every type $X : \mathcal{U}$ and every type universe $\mathcal{V}$, the subset inclusion $\subseteq$ makes $\mathcal{P}_{\mathcal{V}}(X)$ into a poset, where antisymmetry holds by function extensionality and propositional extensionality. Moreover, $\mathcal{P}_{\mathcal{V}}(X)$ has a least element of course: the empty set $\emptyset$. We also claim that $\mathcal{P}_{\mathcal{V}}(X)$ has suprema for all (not necessarily directed) families $\alpha : I \to \mathcal{P}_{\mathcal{V}}(X)$ with $I : \mathcal{V}$. Given such a family α, its least upper bound is given by $\bigcup \alpha :\equiv \lambda x\ .\ \exists_{i:I}\, x \in \alpha_i$, the set-theoretic union, which is well-defined as $(\exists_{i:I}\, x \in \alpha_i) : \mathcal{V}$. It is clear that this is indeed an upper bound for α. And if A is a $\mathcal{V}$-subset of X such that $\alpha_i \subseteq A$ for every $i : I$, then to show that $\bigcup \alpha \subseteq A$ it suffices to construct for every $x : X$, a map $(\Sigma_{i:I}\, x \in \alpha_i) \to (x \in A)$ as $x \in A$ is a subsingleton. But this is easy because we assumed that $\alpha_i \subseteq A$ for every $i : I$. Finally, paying attention to the universe levels we observe that $\mathcal{P}_{\mathcal{V}}(X) : \mathcal{V}\text{-DCPO}_{\mathcal{V}^+\sqcup\mathcal{U},\mathcal{V}\sqcup\mathcal{U}}$. In the case that $X : \mathcal{U} \equiv \mathcal{V}$, we obtain the simpler, locally small $\mathcal{P}_{\mathcal{V}}(X) : \mathcal{V}\text{-DCPO}_{\mathcal{V}^+,\mathcal{V}}$.

Of course, $\Omega_{\mathcal{V}}$ is easily seen to be equivalent to $\mathcal{P}_{\mathcal{V}}(\mathbf{1}_{\mathcal{V}})$, so Example 3.2.15 subsumes Example 3.2.14, but it is instructive to understand Example 3.2.14 first.

Proposition 3.2.16 (ω-completeness)**.** *Every $\mathcal{V}$-dcpo D is* ω-complete, *viz. if we have elements $x_0 \sqsubseteq x_1 \sqsubseteq x_2 \sqsubseteq \dots$ of D, then the supremum of $(x_n)_{n:\mathbb{N}}$ exists in D.*

Proof. Recalling Remark 2.1.2 and using the fact that $\mathbb{N} : \mathcal{U}_0$, the type $\mathrm{lift}_{\mathcal{U}_0,\mathcal{V}}(\mathbb{N})$ is in the universe $\mathcal{V}$ and is equivalent to $\mathbb{N}$. Now $\mathrm{lift}_{\mathcal{U}_0,\mathcal{V}}(\mathbb{N}) \simeq \mathbb{N} \xrightarrow{x_{(-)}} D$ is a directed family as $x_n \sqsubseteq x_{n+1}$ for every natural number n, and it is indexed by a type in $\mathcal{V}$. Hence, it has a least upper bound in D which is the supremum of $(x_n)_{n:\mathbb{N}}$. ☐

3.3 Scott continuous maps

We discuss an appropriate notion of morphism between $\mathcal{V}$-dcpos, namely one that requires preservation of directed suprema and the order (Lemma 3.3.5).

Definition 3.3.1 (Scott continuity)**.** A function $f : D \to E$ between two $\mathcal{V}$-dcpos is *(Scott) continuous* if it preserves directed suprema, i.e. if $I : \mathcal{V}$ and $\alpha : I \to D$ is directed, then $f(\bigsqcup \alpha)$ is the supremum in E of the family $f \circ \alpha$.

Remark 3.3.2. When we speak of a Scott continuous function between D and E, then we will always assume that D and E are both $\mathcal{V}$-dcpos for some arbitrary but fixed type universe $\mathcal{V}$.

Remark 3.3.3. The name "Scott continuous" is due to the fact that such maps are continuous with respect to the so-called Scott topology. We will not discuss the Scott topology in this thesis, but see Chapter 8 for a brief discussion of our work on apartness and the Scott topology in a constructive setting.

Lemma 3.3.4. *Being Scott continuous is a property. In particular, two Scott continuous maps are equal if and only if they are equal as functions.*

Proof. By Theorem 2.3.17 and the fact that being the supremum of a family is a property, cf. Remark 3.2.8. □

Lemma 3.3.5. *If $f : D \to E$ is Scott continuous, then it is* monotone, *i.e. $x \sqsubseteq_D y$ implies $f(x) \sqsubseteq_E f(y)$.*

Proof. Given $x, y : D$ with $x \sqsubseteq y$, consider the directed family $\mathbf{2}_{\mathcal{V}} \xrightarrow{\alpha} D$ defined by $\alpha(0) :\equiv x$ and $\alpha(1) :\equiv y$. Its supremum is y and f must preserve it. Hence, $f(y)$ is an upper bound of $f(\alpha(0)) \equiv f(x)$, so $f(x) \sqsubseteq f(y)$, as we wished to show. □

Lemma 3.3.6. *If $f : D \to E$ is monotone and $\alpha : I \to D$ is directed, then so is $f \circ \alpha$.*

Proof. Since α is directed, I is inhabited, so it remains to prove that $f \circ \alpha$ is semidirected. If we have $i, j : I$, then by directedness of α, there exists $k : I$ such that $\alpha_i, \alpha_j \sqsubseteq \alpha_k$. By monotonicity, we obtain $f(\alpha_i), f(\alpha_j) \sqsubseteq f(\alpha_k)$ as desired. □

Lemma 3.3.7. *A monotone map $f : D \to E$ between $\mathcal{V}$-dcpos is Scott continuous if and only if $f(\bigsqcup \alpha) \sqsubseteq \bigsqcup f \circ \alpha$.*

Note that we are justified in writing $\bigsqcup f \circ \alpha$ because Lemma 3.3.6 tells us that $f \circ \alpha$ is directed by the assumed monotonicity of f.

Proof. The left-to-right implication is immediate. For the converse, note that it only remains to show that $f(\bigsqcup \alpha) \sqsupseteq \bigsqcup f \circ \alpha$. But for this it suffices that $f(\alpha_i) \sqsubseteq f(\bigsqcup \alpha)$ for every $i : I$, which holds as $\bigsqcup \alpha$ is an upper bound of α and f is monotone. □

Remark 3.3.8. In constructive mathematics it is not possible to exhibit a discontinuous function from $\mathbb{N}^{\mathbb{N}}$ to $\mathbb{N}$, because sheaf [TvD88, Chapter 15] and realizability models [vOos08, e.g. Proposition 3.1.6] imply that it is consistent to assume that all such functions are continuous. This does not mean, however, that we cannot exhibit a discontinuous function between dcpos. In fact, the negation map $\neg : \Omega \to \Omega$ is not monotone and hence not continuous. If we were to preclude such examples, then we can no longer work with the full type Ω of all propositions, but instead we must restrict to a subtype of propositions, for example by using dominances [Ros86]. Indeed, this approach is investigated in the context of topos theory in [Pha91; Lon95] and for computability instead of continuity in univalent foundations in [EK17].

Definition 3.3.9 (Strictness)**.** A Scott continuous function $f : D \to E$ between pointed dcpos is *strict* if $f(\bot_D) = \bot_E$.

Lemma 3.3.10. *A poset D is a pointed $\mathcal{V}$-dcpo if and only if it has suprema for all semidirected families indexed by types in $\mathcal{V}$ that we will denote using the $\bigvee$ symbol. In particular, a pointed $\mathcal{V}$-dcpo has suprema of all families indexed by propositions in $\mathcal{V}$. Moreover, if f is a Scott continuous and strict map between pointed $\mathcal{V}$-dcpos, then f preserves suprema of semidirected families.*

Proof. If D is complete with respect to semidirected families indexed by types in $\mathcal{V}$, then it is clearly a $\mathcal{V}$-dcpo and it is pointed because the supremum of the family indexed by the empty type is the least element. Conversely, if D is a pointed $\mathcal{V}$-dcpo and $\alpha : I \to D$ is a semidirected family with $I : \mathcal{V}$, then the family

$$\begin{aligned} \hat{\alpha} : I + \mathbf{1}_{\mathcal{V}} &\to D \\ \mathrm{inl}(i) &\mapsto \alpha_i \\ \mathrm{inr}(\star) &\mapsto \bot \end{aligned}$$

is directed and hence has a supremum in D which is also the least upper bound of α. A pointed $\mathcal{V}$-dcpo must have suprema for all families indexed by propositions in $\mathcal{V}$, because any such family is semidirected. Finally, suppose that $\alpha : I \to D$ is semidirected and that $f : D \to E$ is Scott continuous and strict. Using the $\widehat{(-)}$-construction from above, we see that

$$\begin{aligned} f\left(\bigvee \alpha\right) &\equiv f\left(\bigsqcup \hat{\alpha}\right) \\ &= \bigsqcup f \circ \hat{\alpha} && \text{(by Scott continuity of } f\text{)} \\ &= \bigsqcup \widehat{f \circ \alpha} && \text{(since } f \text{ is strict)} \\ &\equiv \bigvee f \circ \alpha, \end{aligned}$$

finishing the proof. ☐

Proposition 3.3.11.

(i) The identity on any dcpo is Scott continuous.
(ii) For dcpos D and E and $y : E$, the constant map $x \mapsto y : D \to E$ is Scott continuous.
(iii) If $f : D \to E$ and $g : E \to E'$ are Scott continuous, then so is $g \circ f$.

Moreover, if D is pointed, then the identity on D is strict, and if f and g are strict in (iii), then so is $g \circ f$.

Proof. The proofs of (i) and (ii) are obvious. For (iii), let $\alpha : I \to D$ be directed and notice that $g(f(\bigsqcup \alpha)) = g(\bigsqcup f \circ \alpha) = \bigsqcup g \circ f \circ \alpha$ by respectively continuity of f and g. The claims about strictness are also clear. □

Definition 3.3.12 (Isomorphism)**.** A Scott continuous map $f : D \to E$ is an *isomorphism* if we have a Scott continuous inverse $g : E \to D$.

Lemma 3.3.13. *Every $f : D \to E$ isomorphism between pointed dcpos is strict.*

Proof. Let $y : E$ be arbitrary and notice that $\bot_D \sqsubseteq g(y)$ because $\bot_D$ is the least element of D. By monotonicity of f, we get $f(\bot_D) \sqsubseteq f(g(y)) = y$ which shows that $f(\bot_D)$ is the least element of E. □

Definition 3.3.14 (Scott continuous retract)**.** A dcpo D is a *(Scott) continuous retract* of E if we have Scott continuous maps $s : D \to E$ and $r : E \to D$ such that s is a section of r. We denote this situation by $D \underset{r}{\overset{s}{\rightleftarrows}} E$.

Lemma 3.3.15. *If D is a continuous retract of E and E is locally small, then so is D.*

Proof. We claim that $x \sqsubseteq_D y$ and $s(x) \sqsubseteq_E s(y)$ are equivalent, which proves the lemma as E is assumed to be locally small. One direction of the equivalence is given by the fact that s is monotone. In the other direction, assume that $s(x) \sqsubseteq s(y)$ and note that $x = r(s(x)) \sqsubseteq r(s(y)) = y$, as r is monotone and s is a section of r. □

3.4 Lifting

We now turn to constructing pointed $\mathcal{V}$-dcpos from sets. First of all, every discretely ordered set is a $\mathcal{V}$-dcpo, where discretely ordered means that we have $x \sqsubseteq y$ exactly when $x = y$. This is because if $\alpha : I \to X$ is a directed family into a discretely ordered set X, then α has to be constant (by semidirectedness), so α_i is its supremum for any $i : I$. And since directedness includes the condition that the domain is inhabited, it follows that α must have a supremum in X. In fact, ordering X discretely yields the free $\mathcal{V}$-dcpo on the set X in the categorical sense.

With excluded middle, the situation for *pointed* $\mathcal{V}$-dcpos is also very straightforward. Simply order the set X discretely and add a least element, as depicted for $X \equiv \mathbb{N}$ in the Hasse diagram of Proposition 3.4.1. The point of that proposition is to show, by a reduction to the constructive taboo LPO (Definition 2.7.22), that this approach is constructively unsatisfactory. In Chapter 6 we will prove a general constructive no-go theorem (Theorem 6.2.24) showing that there is a nontrivial dcpo with decidable equality if and only if weak excluded middle holds.

Proposition 3.4.1. *If the poset $\mathbb{N}_\bot = (\mathbb{N} + \mathbf{1}, \sqsubseteq)$ with order depicted by the Hasse diagram*

$$\begin{tikzcd}
0 \arrow[drr, dash] & 1 \arrow[dr, dash] & 2 \arrow[d, dash] & 3 \arrow[dl, dash] & \cdots \arrow[dll, dash] \\
& & \bot & &
\end{tikzcd}$$

is ω-complete, then LPO holds. In particular, by Proposition 3.2.16, if it is $\mathcal{U}_0$-directed complete, then LPO holds.

Proof. Let $\alpha : \mathbb{N} \to \mathbf{2}$ be an arbitrary binary sequence. We show that $\exists_{n:\mathbb{N}}\, \alpha_n = 1$ is decidable. Define the family $\beta : \mathbb{N} \to \mathbb{N}_\bot$ by

$$\beta_n :\equiv \begin{cases} \mathrm{inl}(k) & \text{if } k \text{ is the least integer below } n \text{ for which } \alpha_k = 1\text{, and} \\ \mathrm{inr}(\star) & \text{else.} \end{cases}$$

Then β is a chain, so by assumption it has a supremum s in $\mathbb{N}_\bot$. By the induction principle of coproducts, we have $s = \mathrm{inr}(\star)$ or we have $k : \mathbb{N}$ such that $s = \mathrm{inl}(k)$. If the latter holds, then $\alpha_k = 1$, so $\exists_{n:\mathbb{N}}\, \alpha_n = 1$ is decidable. We claim that $s = \mathrm{inr}(\star)$ implies that $\neg(\exists_{n:\mathbb{N}}\, \alpha_n = 1)$. Indeed, assume for a contradiction that $\exists_{n:\mathbb{N}}\, \alpha_n = 1$. Since we are proving a proposition, we may assume to have $n : \mathbb{N}$ with $\alpha_n = 1$. Then, $\beta_n = \mathrm{inl}(k)$ for a natural number $k \leq n$. Since s is the supremum of β we have $\mathrm{inl}(k) = \beta_n \sqsubseteq s = \mathrm{inr}(\star)$, but $\mathrm{inr}(\star)$ is the least element of $\mathbb{N}_\bot$, so by antisymmetry $\mathrm{inl}(k) = \mathrm{inr}(\star)$, which is impossible. □

Our solution to the above will be to work with the lifting monad, sometimes known as the partial map classifier monad from topos theory [Joh77; Ros86; Ros87; Koc91], which has been extended to constructive type theory by Reus and Streicher [RS99] and recently to univalent foundations by Escardó and Knapp [EK17; Kna18]; see the Notes of this chapter for a discussion of additional related work.

Definition 3.4.2 (Lifting, partial element, $\mathcal{L}_\mathcal{V}(X)$; [EK17, Section 2.2])**.** We define the type of *partial elements* of a type $X : \mathcal{U}$ with respect to a universe $\mathcal{V}$ as

$$\mathcal{L}_\mathcal{V}(X) :\equiv \Sigma_{P:\Omega_\mathcal{V}}(P \to X)$$

and we also call it the *lifting* of X with respect to $\mathcal{V}$.

Every (total) element of X gives rise to a partial element of X through the following map, which will be shown to be the unit of the monad later.

Definition 3.4.3 (η_X)**.** The map $\eta_X : X \to \mathcal{L}_\mathcal{V}(X)$ is defined by mapping x to the tuple $(\mathbf{1}_\mathcal{V}, \lambda u \,.\, x)$, where, following Remark 2.4.9, we have omitted the witness that $\mathbf{1}_\mathcal{V}$ is a subsingleton. We sometimes omit the subscript in η_X.

Besides these total elements, the lifting has another distinguished element that will be the least with respect to the order with which we shall equip the lifting.

Definition 3.4.4 ($\perp$)**.** For every type $X : \mathcal{U}$ and universe $\mathcal{V}$, we denote the element $(\mathbf{0}_{\mathcal{V}}, \varphi) : \mathcal{L}_{\mathcal{V}}(X)$ by $\perp$. (Here φ is the unique map from $\mathbf{0}_{\mathcal{V}}$ to X.)

Next we introduce appropriate names for the projections from the type of partial elements.

Definition 3.4.5 (is-defined and value)**.** We write $\text{is-defined} : \mathcal{L}_{\mathcal{V}}(X) \to \Omega_{\mathcal{V}}$ for the first projection and $\text{value} : \Pi_{l:\mathcal{L}_{\mathcal{V}}(X)}(\text{is-defined}(l) \to X)$ for the second projection.

Thus, with this terminology, the element $\star$ witnesses that $\eta(x)$ is defined with value x for every $x : X$, while $\perp$ is not defined because $\text{is-defined}(\perp)$ is the empty type.

Excluded middle says exactly that such elements are the only elements of the lifting of a type X, as the following proposition shows. Thus, the lifting generalises the classical construction of adding a new element.

Proposition 3.4.6 ([EK17, Section 2.2])**.** *The map $X + \mathbf{1} \xrightarrow{[\eta, \text{const}_\perp]} \mathcal{L}_{\mathcal{V}}(X)$ is an equivalence for every type $X : \mathcal{U}$ if and only if excluded middle in $\mathcal{V}$ holds.*

Proof. By the proof of Lemma 2.7.21(iv), excluded middle in $\mathcal{V}$ is equivalent to the map $[\text{const}_{\mathbf{0}}, \text{const}_{\mathbf{1}}] : \mathbf{2}_{\mathcal{V}} \to \Omega_{\mathcal{V}}$ being an equivalence. But if that map is an equivalence, then it follows that the map $[\eta, \text{const}_\perp] : X + \mathbf{1} \to \mathcal{L}_{\mathcal{V}}(X)$ is also an equivalence for every type X. Conversely, we can take $X :\equiv \mathbf{1}_{\mathcal{V}}$ to see that $[\text{const}_{\mathbf{0}}, \text{const}_{\mathbf{1}}] : \mathbf{2}_{\mathcal{V}} \to \Omega_{\mathcal{V}}$ must be an equivalence. □

Lemma 3.4.7. *Two partial elements $l, m : \mathcal{L}_{\mathcal{V}}(X)$ of a type X are equal if and only if we have* $\text{is-defined}(l) \iff \text{is-defined}(m)$ *and the diagram*

$$\begin{tikzcd}
\text{is-defined}(m) \arrow[rr, "\text{value}(m)"] \arrow[dr] & & X \\
& \text{is-defined}(l) \arrow[ur, "\text{value}(l)"'] &
\end{tikzcd}$$

commutes.

Proof. By Lemma 2.2.5 we have

$$(l = m) \simeq \left(\Sigma_{e:\text{is-defined}(l)=\text{is-defined}(m)}\ \text{transport}^{\lambda P.P \to X}(e, \text{value}(l)) = \text{value}(m)\right)$$

By path induction on e we can prove that

$$\text{transport}^{\lambda P.P \to X}(e, \text{value}(l)) = \text{value}(l) \circ \tilde{e}^{-1},$$

where $\tilde{e}$ is the equivalence $\text{is-defined}(l) \simeq \text{is-defined}(m)$ induced by e. Hence, using function extensionality and propositional extensionality, the right hand side of the equivalence given above is logically equivalent to

$$\Sigma_{(e_1,e_2):\text{is-defined}(l)\leftrightarrow\text{is-defined}(m)}\ \text{value}(l) \circ e_2 \sim \text{value}(m),$$

as desired. □

Remark 3.4.8. It is possible to promote the logical equivalence of Lemma 3.4.7 to an equivalence of types using univalence and a generalised *structure identity principle* [Esc19b, Section 3.33], as done in [Esc21, Lemma 44] and [Esc+, `Lifting.IdentityViaSIP`]. But the above logical equivalence will suffice.

Theorem 3.4.9 (Lifting monad, Kleisli extension, $f^\#$; [EK17, Section 2.2]). *The lifting is a monad with unit η. That is, for every map $f : X \to \mathcal{L}_{\mathcal{V}}(Y)$ we have a map $f^\# : \mathcal{L}_{\mathcal{V}}(X) \to \mathcal{L}_{\mathcal{V}}(Y)$, the* Kleisli extension *of f, such that*

(i) $\eta_X^\# \sim \mathrm{id}_{\mathcal{L}_{\mathcal{V}}(X)}$ for every type X,

(ii) $f^\# \circ \eta_X \sim f$ for every map $f : X \to \mathcal{L}_{\mathcal{V}}(Y)$, and

(iii) $(g^\# \circ f)^\# \sim g^\# \circ f^\#$ for every two maps $f : X \to \mathcal{L}_{\mathcal{V}}(Y)$ and $g : Y \to \mathcal{L}_{\mathcal{V}}(Z)$.

Proof. Given $f : X \to \mathcal{L}_{\mathcal{V}}(Y)$, we define

$$\begin{aligned} f^\# &: \mathcal{L}_{\mathcal{V}}(X) \to \mathcal{L}_{\mathcal{V}}(Y) \\ &(P, \varphi) \mapsto \left(\Sigma_{p:P}\, \text{is-defined}(f(\varphi(p))), \psi\right), \end{aligned}$$

where $\psi(p, q) :\equiv \text{value}(f(\varphi(p)), q)$.
Now for the proof of (i): Let $(P, \varphi) : \mathcal{L}_{\mathcal{V}}(X)$ be arbitrary and we calculate that

$$\begin{aligned} \eta^\#(P, \varphi) &\equiv \left(\Sigma_{p:P}\, \text{is-defined}(\eta(\varphi(p))), \lambda\,(p, q)\,.\, \text{value}(\eta(\varphi(p)), q)\right) \\ &\equiv (P \times \mathbf{1}, \lambda\,(p, q)\,.\, \varphi(p)) \\ &= (P, \varphi), \end{aligned}$$

where the final equality is seen to hold using Lemma 3.4.7. For (ii), let $x : X$ and $f : X \to \mathcal{L}_{\mathcal{V}}(Y)$ be arbitrary and observe that

$$\begin{aligned} f^\#(\eta(x)) &\equiv f^\#(\mathbf{1}, \lambda\, u\,.\, x) \\ &\equiv (\mathbf{1} \times \text{is-defined}(f(x)), \lambda\,(u, p)\,.\, \text{value}(f(x), p)) \\ &= (\text{is-defined}(f(x)), \lambda\, p\,.\, \text{value}(f(x), p)) \\ &\equiv f(x) \end{aligned}$$

where the penultimate equality is another easy application of Lemma 3.4.7. We see that these proofs amount to the fact that $\mathbf{1}$ is the unit for taking the product of types. For (iii) the proof amounts to the associativity of Σ. □

Remark 3.4.10. It should be noted that if $X : \mathcal{U}$, then $\mathcal{L}_{\mathcal{V}}(X) : \mathcal{V}^+ \sqcup \mathcal{U}$, so in general the lifting is a monad "across universes". But this increase in universes does not hinder us in stating and proving the monad laws and using them in later proofs.

Remark 3.4.11. The equalities of Theorem 3.4.9 do not include any coherence conditions which may be needed when X is not a set but a higher type. We will restrict to the lifting of sets, but the more general case is considered in [Esc21] where the coherence conditions are not needed for its goals either.

Definition 3.4.12 (Lifting functor, $\mathcal{L}_{\mathcal{V}}(f)$)**.** The functorial action of the lifting could be defined from the unit and Kleisli extension as $\mathcal{L}_{\mathcal{V}}(f) :\equiv (\eta_Y \circ f)^{\#}$ for $f : X \to Y$. But it is equivalent and easier to define $\mathcal{L}_{\mathcal{V}}(f)$ directly by post-composition:

$$\mathcal{L}_{\mathcal{V}}(f)(P, \varphi) :\equiv (P, f \circ \varphi).$$

We now work towards showing that $\mathcal{L}_{\mathcal{V}}(X)$ is the free pointed $\mathcal{V}$-dcpo on a set X.

Proposition 3.4.13. *The relation $\sqsubseteq : \mathcal{L}_{\mathcal{V}}(X) \to \mathcal{L}_{\mathcal{V}}(X) \to \mathcal{V}^+ \sqcup \mathcal{U}$ given by*

$$l \sqsubseteq m :\equiv \text{is-defined}(l) \to l = m$$

is a partial order on $\mathcal{L}_{\mathcal{V}}(X)$ for every set $X : \mathcal{U}$. Moreover, it is equivalent to the more verbose relation

$$(P, \varphi) \sqsubseteq' (Q, \psi) :\equiv \Sigma_{f : P \to Q} (\varphi \sim \psi \circ f)$$

that is valued in $\mathcal{V} \sqcup \mathcal{U}$.

Proof. Note that $\sqsubseteq$ is subsingleton-valued because X is assumed to be a set. The other properties follow using Lemma 3.4.7. ☐

In light of Remark 3.2.10, we carefully keep track of the universe parameters of the lifting in the following proposition.

Proposition 3.4.14 (cf. [EK17, Theorem 1])**.** *For a set $X : \mathcal{U}$, the lifting $\mathcal{L}_{\mathcal{V}}(X)$ ordered as in Proposition 3.4.13 is a pointed $\mathcal{V}$-dcpo. In general, $\mathcal{L}_{\mathcal{V}}(X) : \mathcal{V}\text{-DCPO}_{\mathcal{V}^+ \sqcup \mathcal{U}, \mathcal{V}^+ \sqcup \mathcal{U}}$, but if $X : \mathcal{V}$, then $\mathcal{L}_{\mathcal{V}}(X)$ is locally small.*

Proof. By Proposition 3.4.13 we have a poset and it is clear that $\bot$ from Definition 3.4.4 is its least element. Now let $\left(Q_{(-)}, \varphi_{(-)}\right) : I \to \mathcal{L}_{\mathcal{V}}(X)$ be a directed family with $I : \mathcal{V}$. We claim that the map $\Sigma_{i:I} Q_i \xrightarrow{(i,q) \mapsto \varphi_i(q)} X$ is constant. Indeed, given $i, j : I$ with $p : Q_i$ and $q : Q_j$, there exists $k : I$ such that $(Q_i, \varphi_i), \left(Q_j, \varphi_j\right) \sqsubseteq (Q_k, \varphi_k)$ by directedness of the family. But by definition of the order and the elements $p : Q_i$ and $q : Q_j$, this implies that $(Q_i, \varphi_i) = \left(Q_j, \varphi_j\right) = (Q_k, \varphi_k)$ which in particular tells us that $\varphi_i(p) = \varphi_j(q)$. Hence, by Theorem 2.6.9, we have a (dashed) map ψ making the diagram

$$\begin{tikzcd}
\Sigma_{i:I} Q_i \arrow[rr, "(i,q) \mapsto \varphi_i(q)"] \arrow[dr, "|-|"'] & & X \\
& \exists_{i:I} Q_i \arrow[ur, dashed, "\psi"'] &
\end{tikzcd}$$

commute. We claim that $(\exists_{i:I} Q_i, \psi)$ is the least upper bound of the family. To see that it is an upper bound, let $j : I$ be arbitrary. By the commutative diagram and Proposition 3.4.13 we see that $\left(Q_j, \varphi_j\right) \sqsubseteq (\exists_{i:I} Q_i, \psi)$, as desired. Moreover, if (P, ρ) is an upper bound for the family, then $(Q_i, \varphi_i) = (P, \rho)$ for all $i : I$ such that Q_i holds. Hence, $(\exists_{i:I} Q_i, \psi) \sqsubseteq (P, \rho)$, as desired. Finally, local smallness in the case that X is a type in $\mathcal{V}$ follows from Proposition 3.4.13. ☐

Proposition 3.4.15. *The Kleisli extension $f^\# : \mathcal{L}_{\mathcal{V}}(X) \to \mathcal{L}_{\mathcal{V}}(Y)$ is Scott continuous for any map $f : X \to \mathcal{L}_{\mathcal{V}}(Y)$.*

Proof. It is straightforward to prove that $f^\#$ is monotone. Hence, it remains to prove that $f^\#(\bigsqcup \alpha) \sqsubseteq \bigsqcup f^\# \circ \alpha$ for every directed family $\alpha : I \to \mathcal{L}_{\mathcal{V}}(X)$. So suppose that $f^\#(\bigsqcup \alpha)$ is defined. Then we have to show that it equals $\bigsqcup f^\# \circ \alpha$. By our assumption and definition of $f^\#$ we get that $\bigsqcup \alpha$ is defined too. By the definition of suprema in the lifting and because we are proving a proposition, we may assume to have $i : I$ such that α_i is defined. But since $\alpha_i \sqsubseteq \bigsqcup \alpha$, we get $\alpha_i = \bigsqcup \alpha$ and hence, $f^\#(\alpha_i) = f^\#(\bigsqcup \alpha)$. Finally, $f^\#(\alpha_i) \sqsubseteq \bigsqcup f^\# \circ \alpha$, but by assumption $f^\#(\bigsqcup \alpha)$ is defined and hence so is $f^\#(\alpha_i)$ which implies $f^\#(\bigsqcup \alpha) = f^\#(\alpha_i) = \bigsqcup f^\# \circ \alpha$, as desired. □

Recall from Lemma 3.3.10 that pointed $\mathcal{V}$-dcpos have suprema of families indexed by propositions in $\mathcal{V}$. We make use of this fact in the following lemma.

Lemma 3.4.16. *For a set X, every partial element $(P, \varphi) : \mathcal{L}_{\mathcal{V}}(X)$ is equal to supremum $\bigvee_{p:P} \eta_X(\varphi(p))$.*

Proof. Note that if $p : P$, then $(P, \varphi) = \eta_X(\varphi(p))$, so that the lemma follows from antisymmetry. □

The lifting $\mathcal{L}_{\mathcal{V}}(X)$ gives the free pointed $\mathcal{V}$-dcpo on a set X. Keeping track of universes, it holds in the following generality:

Theorem 3.4.17. *If $X : \mathcal{U}$ is a set, then for every pointed $\mathcal{V}$-dcpo $D : \mathcal{V}\text{-DCPO}_{\mathcal{U}',\mathcal{T}'}$ and function $f : X \to D$, there is a unique strict and continuous function $\bar{f} : \mathcal{L}_{\mathcal{V}}(X) \to D$ making the diagram*

$$\begin{tikzcd} X \arrow[rr, "f"] \arrow[dr, "\eta_X"'] & & D \\ & \mathcal{L}_{\mathcal{V}}(X) \arrow[ur, dashed, "\bar{f}"'] & \end{tikzcd}$$

commute.

Proof. We define $\bar{f} : \mathcal{L}_{\mathcal{V}}(X) \to D$ by $(P, \varphi) \mapsto \bigvee_{p:P} f(\varphi(p))$, which is well-defined by Lemma 3.3.10 and easily seen to be strict and continuous. For uniqueness, suppose that we have $g : \mathcal{L}_{\mathcal{V}}(X) \to D$ strict and continuous such that $g \circ \eta_X = f$ and let (P, φ) be an arbitrary element of $\mathcal{L}_{\mathcal{V}}(X)$. Then,

$$\begin{aligned} g(P, \varphi) &= g\Big(\textstyle\bigvee_{p:P} \eta_X(\varphi(p))\Big) && \text{(by Lemma 3.4.16)} \\ &= \textstyle\bigvee_{p:P} g(\eta_X(\varphi(p))) && \text{(by Lemma 3.3.10 and strictness and continuity of } g\text{)} \\ &= \textstyle\bigvee_{p:P} f(\varphi(p)) && \text{(by assumption on } g\text{)} \\ &\equiv \bar{f}(P, \varphi), \end{aligned}$$

as desired. □

The proof tells us that there is yet another way in which the lifting is a free construction, namely as the free subsingleton-complete poset. What is noteworthy about this is that freely adding subsingleton suprema automatically gives all directed suprema.

Definition 3.4.18 (Subsingleton completeness)**.** A poset P is *subsingleton complete* with respect to a type universe $\mathcal{V}$ if it has suprema for all families indexed by a subsingleton in $\mathcal{V}$.

The lifting $\mathcal{L}_{\mathcal{V}}(X)$ gives the free $\mathcal{V}$-subsingleton complete poset on a set X. Keeping track of universes, it holds in the following generality:

Theorem 3.4.19. *If $X : \mathcal{U}$ is a set, then for every $\mathcal{V}$-subsingleton complete poset P (with carrier and order taking values in arbitrary, possibly distinct, universes) and function $f : X \to P$, there exists a unique monotone $\bar{f} : \mathcal{L}_{\mathcal{V}}(X) \to P$ preserving all suprema indexed by propositions in $\mathcal{V}$ making the diagram*

$$\begin{tikzcd} X \arrow[rr, "f"] \arrow[dr, "\eta_X"'] & & P \\ & \mathcal{L}_{\mathcal{V}}(X) \arrow[ur, dashed, "\bar{f}"'] & \end{tikzcd}$$

commute.

Proof. Similar to the proof of Theorem 3.4.17. ☐

Finally, we consider a variation of Proposition 3.4.13 which allows us to freely add a least element to a $\mathcal{V}$-dcpo instead of just a set.

Proposition 3.4.20. *For a poset D whose order takes values in $\mathcal{T}$, the binary relation $\sqsubseteq : \mathcal{L}_{\mathcal{V}}(D) \to \mathcal{L}_{\mathcal{V}}(D) \to \mathcal{V} \sqcup \mathcal{T}$ given by*

$$(P, \varphi) \sqsubseteq (Q, \psi) :\equiv \Sigma_{f : P \to Q}\big(\Pi_{p:P}(\varphi(p) \sqsubseteq_D \psi(f(p)))\big)$$

is a partial order on $\mathcal{L}_{\mathcal{V}}(D)$.

Proof. Similar to Proposition 3.4.13, but using that $\sqsubseteq_D$ is reflexive, transitive and antisymmetric. ☐

Proposition 3.4.21. *For a dcpo $D : \mathcal{V}\text{-DCPO}_{\mathcal{U},\mathcal{T}}$, the lifting $\mathcal{L}_{\mathcal{V}}(D)$ ordered as in Proposition 3.4.20 is a pointed $\mathcal{V}$-dcpo. In general, $\mathcal{L}_{\mathcal{V}}(D) : \mathcal{V}\text{-DCPO}_{\mathcal{V}^+ \sqcup \mathcal{U}, \mathcal{V} \sqcup \mathcal{T}}$, but if D is locally small, then so is $\mathcal{L}_{\mathcal{V}}(D)$.*

Proof. The element $\perp$ from Definition 3.4.4 is still the least element with respect to the new order. If $\alpha : I \to \mathcal{L}_{\mathcal{V}}(D)$ is directed, then, writing $(Q_i, \varphi_i) :\equiv \alpha_i$, we consider $\Phi : (\Sigma_{i:I} Q_i) \to D$ given by $(i, q) \mapsto \varphi_i(q)$. This family is semidirected, for if we have $i, j : I$ with $p : Q_i$ and $q : Q_j$, then there exists $k : I$ such that $\alpha_i, \alpha_j \sqsubseteq \alpha_k$ in $\mathcal{L}_{\mathcal{V}}(D)$ by directedness of α, which implies that $\Phi(i, p) \sqsubseteq \Phi(j, q)$ in D. Thus, if we know that $\exists_{i:I} Q_i$, then the family Φ is directed and must have a supremum in D. Hence we have a partial element $(\exists_{i:I} Q_i, \psi) : \mathcal{L}_{\mathcal{V}}(D)$ where ψ takes the witness that the domain of Φ is inhabited to the directed supremum $\bigsqcup \Phi$ in D. It is not hard to verify that this partial element is the least upper bound of α in $\mathcal{L}_{\mathcal{V}}(D)$, completing the proof. □

The lifting $\mathcal{L}_{\mathcal{V}}(D)$ with the partial order of Proposition 3.4.20 gives the free *pointed* $\mathcal{V}$-dcpo on a $\mathcal{V}$-dcpo D. Keeping track of universes, it holds in the following generality:

Theorem 3.4.22. *If $D : \mathcal{V}\text{-DCPO}_{\mathcal{U},\mathcal{T}}$ is a $\mathcal{V}$-dcpo, then for every pointed $\mathcal{V}$-dcpo $E : \mathcal{V}\text{-DCPO}_{\mathcal{U}',\mathcal{T}'}$ and continuous function $f : D \to E$, there is a unique strict continuous function $\bar{f} : \mathcal{L}_{\mathcal{V}}(D) \to E$ making the diagram*

$$\begin{tikzcd}
D \arrow[rr, "f"] \arrow[dr, "\eta_D"'] & & E \\
& \mathcal{L}_{\mathcal{V}}(D) \arrow[ur, dashed, "\bar{f}"'] &
\end{tikzcd}$$

commute.

Proof. Similar to the proof of Theorem 3.4.17. □

Notice how Theorem 3.4.22 generalises Theorem 3.4.17 as any set can be viewed as a discretely ordered $\mathcal{V}$-dcpo.

3.5 Products and exponentials

We describe two constructions of $\mathcal{V}$-dcpos, namely products and exponentials. Exponentials will be crucial in the Scott model of PCF, as discussed in Section 5.2. Products are not needed for this purpose as we will work with the combinatory version of PCF. However, product allows us to state the universal property of the exponential (Proposition 3.5.8). Moreover, products are also needed when extending the Scott model to account for a version of PCF with variables and λ-abstraction, see the Notes for this chapter.

Definition 3.5.1 (Product of (pointed) dcpos, $D_1 \times D_2$)**.** The *product* of two $\mathcal{V}$-dcpos D_1 and D_2 is given by the $\mathcal{V}$-dcpo $D_1 \times D_2$ defined as follows. Its carrier is the cartesian product of the carriers of D_1 and D_2. The order is given componentwise, i.e. $(x, y) \sqsubseteq_{D_1 \times D_2} (x', y')$ if $x \sqsubseteq_{D_1} y$ and $y \sqsubseteq_{D_2} y'$. Accordingly directed suprema are also given componentwise. That is, given a directed family $\alpha : I \to D_1 \times D_2$, one quickly verifies that the families $\mathrm{pr}_1 \circ \alpha$ and $\mathrm{pr}_2 \circ \alpha$ are also directed. We then define the supremum of α as $(\bigsqcup \mathrm{pr}_1 \circ \alpha, \bigsqcup \mathrm{pr}_2 \circ \alpha)$. Moreover, if D and E are pointed, then so is $D \times E$ by taking the least elements in both components.

Remark 3.5.2. Notice that if $D_1 : \mathcal{V}\text{-DCPO}_{\mathcal{U},\mathcal{T}}$ and $D_2 : \mathcal{V}\text{-DCPO}_{\mathcal{U}',\mathcal{T}'}$, then for their product we have $D_1 \times D_2 : \mathcal{V}\text{-DCPO}_{\mathcal{U}\sqcup\mathcal{U}',\mathcal{T}\sqcup\mathcal{T}'}$, which simplifies to $\mathcal{V}\text{-DCPO}_{\mathcal{U},\mathcal{T}}$ when $\mathcal{U}' \equiv \mathcal{U}$ and $\mathcal{T}' \equiv \mathcal{T}$.

Proposition 3.5.3. *The product defined above satisfies the appropriate universal property: the projections* $\mathrm{pr}_1 : D_1 \times D_2 \to D_1$ *and* $\mathrm{pr}_2 : D_1 \times D_2 \to D_2$ *are Scott continuous and if* $f : E \to D_1$ *and* $g : E \to D_2$ *are Scott continuous functions from a* $\mathcal{V}$*-dcpo* E*, then there is a unique Scott continuous map* $k \colon E \to D_1 \times D_2$ *such that the diagram*

$$\begin{tikzcd}
& D_1 \times D_2 \arrow[dl, "\mathrm{pr}_1"'] \arrow[dr, "\mathrm{pr}_2"] & \\
D_1 & & D_2 \\
& E \arrow[ul, "f"] \arrow[ur, "g"'] \arrow[uu, dashed, "k"] &
\end{tikzcd}$$

commutes.

Proof. The projections are Scott continuous by definition of directed suprema in $D_1 \times D_2$. Moreover, if $f \colon E \to D_1$ and $g \colon E \to D_2$ are Scott continuous maps, then we see that we have no choice but to define $k \colon E \to D_1 \times D_2$ by $e \mapsto (f(e), g(e))$. Moreover, this assignment is Scott continuous, because for a directed family $\alpha : I \to E$, we have $k(\bigsqcup \alpha) \equiv (f(\bigsqcup \alpha), g(\bigsqcup \alpha)) = (\bigsqcup f \circ \alpha, \bigsqcup g \circ \alpha) \equiv \bigsqcup k \circ \alpha$ by Scott continuity of f and g and the definition of directed suprema in $D_1 \times D_2$. ☐

Lemma 3.5.4. *A map* $f \colon D_1 \times D_2 \to E$ *is Scott continuous if and only if the maps* $f(x, -) : D_2 \to E$ *and* $f(-, y) : D_1 \to E$ *are Scott continuous for every* $x : D_1$ *and* $y : D_2$.

Proof. Suppose first that $f \colon D_1 \times D_2 \to E$ is Scott continuous and let $x : D_1$ be arbitrary. If $\alpha : I \to D_2$ is a directed family, then $f(x, \bigsqcup \alpha) = f(\bigsqcup \alpha_x)$, where $\alpha_x : I \to D_1 \times D_2$ is the directed family given by $i \mapsto (x, \alpha(i))$. But f is Scott continuous, so $\bigsqcup_{i:I} f(x, \alpha(i)) = \bigsqcup_{i:I} f(\alpha_x(i)) = f(\bigsqcup \alpha_x(i)) = f(x, \bigsqcup \alpha)$, as desired. Continuity of $f(-, y)$ is proved similarly of course.
Conversely, suppose that the conditions in the lemma hold and let $\alpha : I \to D_1 \times D_2$ be directed. We need to show that $f(\bigsqcup \alpha) \equiv f(\bigsqcup \alpha_1, \bigsqcup \alpha_2)$ is the least upper bound of $\bigsqcup f \circ \alpha$, where $\alpha_1 :\equiv \mathrm{pr}_1 \circ \alpha$ and $\alpha_2 :\equiv \mathrm{pr}_2 \circ \alpha$. To see that it is indeed an upper bound, assume that $i : I$ and observe that

$$f(\alpha(i)) \equiv f(\alpha_1(i), \alpha_2(i)) \sqsubseteq f(\alpha_1(i), \textstyle\bigsqcup \alpha_2) \sqsubseteq f(\bigsqcup \alpha_1, \bigsqcup \alpha_2),$$

by monotonicity of $f(\alpha_1(i), -)$ and $f(-, \bigsqcup \alpha_2)$. To see that it is least, suppose that $y \sqsupseteq f(\alpha(i))$ for every $i : I$. By Scott continuity of $f(-, \bigsqcup \alpha_2)$ it is sufficient to prove that $f(\alpha_1(i), \bigsqcup \alpha_2)$ for every $i : I$. So let $i : I$ be arbitrary. By Scott continuity of $f(\alpha_1(i), -)$ it suffices to prove $f(\alpha_1(i), \alpha_2(j)) \sqsubseteq y$ for every $j : I$. So let $j : I$ be arbitrary. By directedness of α, there exists $k : I$ such that $\alpha(i), \alpha(j) \sqsubseteq \alpha(k)$. Hence, $f(\alpha_1(i), \alpha_2(j)) \sqsubseteq f(\alpha(k)) \sqsubseteq y$, as desired. ☐

Definition 3.5.5 (Exponential of (pointed) dcpos, E^D)**.** The *exponential* of two $\mathcal{V}$-dcpos D and E is given by the poset E^D defined as follows. Its carrier is the type of Scott continuous functions from D to E. The order is given pointwise, i.e. $f \sqsubseteq_{E^D} g$ holds if $f(x) \sqsubseteq_E g(x)$ for every $x : D$. Notice that if E is pointed, then so is E^D with least element given the constant function $\lambda\, x : D\,.\, \bot_E$ which is Scott continuous by Proposition 3.3.11(ii).

Note that the exponential E^D is a priori not locally small even if E is because the partial order quantifies over all elements of D. But if D is continuous (a notion that we will study in detail in Chapter 4) then E^D will be locally small when E is (Proposition 4.7.11).

Proposition 3.5.6. *The exponential E^D of two $\mathcal{V}$-dcpos D and E is $\mathcal{V}$-directed complete.*

Proof. Since the partial order is given pointwise, we expect directed suprema to be calculated pointwise too. Explicitly, given a directed family $\alpha : I \to E^D$, we verify that for every $x : D$, the family $\alpha_x : I \to E$ defined by $i \mapsto \alpha_i(x)$ is also directed. Indeed, if we have $i, j : I$, then there exists $k : I$ such that $\alpha_i, \alpha_j \sqsubseteq \alpha_k$. Hence, for arbitrary $x : D$, we have $\alpha_i(x), \alpha_j(x) \sqsubseteq \alpha_k(x)$, which shows that α_x is directed. Because the order is pointwise, it is clear that the function $\lambda\, x : D\,.\, \bigsqcup \alpha_x$ is the least upper bound of α, but we must also check that this function is Scott continuous. We employ Lemma 3.3.7 for this, so we first check that the function is monotone. Indeed if $x \sqsubseteq y$ in D, then $\alpha_i(x) \sqsubseteq \alpha_i(y)$ for every $i : I$ as Scott continuous functions are monotone. Hence, $\bigsqcup \alpha_x \sqsubseteq \bigsqcup \alpha_y$ in this case. Now let $\beta : J \to D$ be directed. We have to prove that $\bigsqcup_{i:I} \alpha_i(\bigsqcup \beta) \sqsubseteq \bigsqcup_{j:J} \bigsqcup_{i:I} \alpha_i(\beta_j)$, for which it is enough to know that $\alpha_i(\bigsqcup \beta) \sqsubseteq \bigsqcup_{j:J} \bigsqcup_{i:I} \alpha_i(\beta_j)$ for every $i : I$. But this is clear as $\alpha_i(\bigsqcup \beta) = \bigsqcup_{j:J} \alpha_i(\beta_j)$ by Scott continuity of each α_i. $\square$

Remark 3.5.7. Recall from Remark 3.2.10 that it is necessary to carefully keep track of the universe parameters of the exponential. In general, the universe levels of E^D can be quite large and complicated. For if $D : \mathcal{V}\text{-DCPO}_{\mathcal{U},\mathcal{T}}$ and $E : \mathcal{V}\text{-DCPO}_{\mathcal{U}',\mathcal{T}'}$, then

$$E^D : \mathcal{V}\text{-DCPO}_{\mathcal{V}^+ \sqcup \mathcal{U} \sqcup \mathcal{T} \sqcup \mathcal{U}' \sqcup \mathcal{T}', \mathcal{U} \sqcup \mathcal{T}'}\,.$$

Even if $\mathcal{V} = \mathcal{U} \equiv \mathcal{T} \equiv \mathcal{U}' \equiv \mathcal{T}'$, the carrier of E^D still lives in the larger universe $\mathcal{V}^+$, because the type expressing Scott continuity for $\mathcal{V}$-dcpos quantifies over all types in $\mathcal{V}$. Actually, the scenario where $\mathcal{U} = \mathcal{U}' = \mathcal{V}$ cannot happen in a predicative setting unless D and E are trivial, in a sense made precise in Chapter 6.
Even so, in many applications such as those in Chapter 5, if we take $\mathcal{V} \equiv \mathcal{U}_0$ and all other parameters to be $\mathcal{U} \equiv \mathcal{T} \equiv \mathcal{U}' \equiv \mathcal{T}' \equiv \mathcal{U}_1$, then the situation is much simpler and D, E and the exponential E^D are all elements of $\mathcal{U}_0\text{-DCPO}_{\mathcal{U}_1,\mathcal{U}_1}$ with all of them being locally small (remember that this is defined up to equivalence). This turns out to be a very favourable situation for both the Scott model of PCF and Scott's D_∞ model of the untyped λ-calculus.

In the proposition below we can have $D : \mathcal{V}\text{-DCPO}_{\mathcal{U},\mathcal{T}}$ and $E : \mathcal{V}\text{-DCPO}_{\mathcal{U}',\mathcal{T}'}$ for arbitrary universes $\mathcal{U}, \mathcal{T}, \mathcal{U}'$ and $\mathcal{T}'$. In particular, the universe parameters of D and E, apart from the universe of indexing types, need to be the same.

Proposition 3.5.8. *The exponential defined above satisfies the appropriate universal property: the* evaluation map ev $: E^D \times D \to E, (g, x) \mapsto g(x)$ *is Scott continuous and if $f : D' \times D \to E$ is a Scott continuous function, then there is a unique Scott continuous map $\bar{f} : D' \to E^D$ such that the diagram*

$$\begin{tikzcd} D' \times D \arrow[d, dashed, "\bar{f} \times \mathrm{id}_D"'] \arrow[dr, "f"] & \\ E^D \times D \arrow[r, "\mathrm{ev}"'] & E \end{tikzcd}$$

commutes.

Proof. We use Lemma 3.5.4 to prove that ev is Scott continuous: It is continuous in the second argument, because the first argument is a Scott continuous function, and it is continuous in the first argument, because suprema in the exponential are calculated pointwise. From the diagram we see that we have no choice but to define $\bar{f}$ as $y \mapsto \lambda x \,.\, f(y, x)$. It remains to prove that $\bar{f}(y)$ is Scott continuous for every $y : D'$ and that this assignment itself defines a Scott continuous function $D' \to E^D$. For the former, note that $\bar{f}(y) :\equiv f(y, -)$ is indeed Scott continuous by Lemma 3.5.4. For the latter, note that if $\alpha : I \to D'$ is directed, then

$$\bar{f}(\bigsqcup \alpha) \equiv \lambda x \,.\, f(\bigsqcup \alpha, x) = \lambda x \,.\, \bigsqcup_{i:I} f(\alpha_i, x) \equiv \bigsqcup_{i:I} (\lambda x \,.\, f(\alpha_i, x))$$

by Scott continuity of f and the fact that suprema are calculated pointwise in the exponential. Thus, $\bar{f}$ is Scott continuous, completing the proof. $\square$

The following theorem lies at the heart of the Scott model of PCF that we will study in Section 5.2.

Theorem 3.5.9 (Least fixed point, μ). *Every Scott continuous endomap f on a pointed $\mathcal{V}$-dcpo D has a least fixed point given by*

$$\mu(f) :\equiv \bigsqcup_{n:\mathbb{N}} f^n(\bot).$$

Specifically, the following two conditions hold:

(i) *$f(\mu(f)) = \mu(f)$, and*

(ii) *for every $x : D$, if $f(x) \sqsubseteq x$, then $\mu(f) \sqsubseteq x$.*

Moreover, the assignment $f \mapsto \mu(f)$ defines a Scott continuous map $D^D \to D$.

Proof. We follow the proof given in [AJ94, Theorem 2.1.19] and first establish (ii). Suppose that $f(x) \sqsubseteq x$. To show that $\mu(f) \sqsubseteq x$, it suffices to prove that $f^n(\bot) \sqsubseteq x$ for every $n : \mathbb{N}$. But this follows easily by induction on n and the fact that f is monotone. For (i), first notice that

$$f(\mu(f)) \equiv f(\bigsqcup_{n:\mathbb{N}} f^n(\bot)) = \bigsqcup_{n:\mathbb{N}} f^{n+1}(\bot) \sqsubseteq \mu(f) \qquad (\dagger)$$

by Scott continuity of f, proving one of the inequalities. But f is monotone, so (†) yields $f(f(\mu(f))) \sqsubseteq f(\mu(f))$, which by (ii) implies $\mu(f) \sqsubseteq f(\mu(f))$, so that $f(\mu(f)) = \mu(f)$ by antisymmetry as we set out to prove. To see that the assignment $f \mapsto \mu(f)$ is continuous we will reconstruct it as the least upper bound of a family in the exponential $D^{(D^D)}$. First define for every natural number $n : \mathbb{N}$, the function

$$\begin{aligned}\mathrm{iter}_n : D^D &\to D \\ f &\mapsto f^n(\bot)\end{aligned}$$

Observe that iter_n can be factored as $D^D \xrightarrow{f \mapsto f^n} D^D \xrightarrow{\text{evaluate at } \bot} D$. By induction on n and Proposition 3.3.11 the first map is seen to be continuous, while the second is continuous by Proposition 3.5.8. Hence, the composite, iter_n is continuous for every $n : \mathbb{N}$ by Proposition 3.3.11. Thus, each iter_n is an element of $D^{(D^D)}$. Moreover, the assignment $n \mapsto \mathrm{iter}_n$ is directed in $D^{(D^D)}$ because if $n \leq m$, then $\mathrm{iter}_n(f) \equiv f^n(\bot) \sqsubseteq f^m(\bot) \equiv \mathrm{iter}_m(f)$ for every Scott continuous $f : D \to D$. Hence, we can take the supremum of $(\mathrm{iter})_{n:\mathbb{N}}$ in $D^{(D^D)}$ which is Scott continuous by construction. But suprema are calculated pointwise, so we can compute that $(\bigsqcup \mathrm{iter})(f) \equiv \bigsqcup_{n:\mathbb{N}} f^n(\bot)$, establishing the continuity of $f \mapsto \mu(f)$ and completing the proof. □

In the Scott model of PCF (Section 5.2), the μ operation is used to model general recursion in the programming language PCF. The equation $f(\mu(f)) = \mu(f)$ may be regarded as the unfolding of a recursive definition, while the least element $\bot$ represents nontermination.

3.6 Bilimits

Recall that in a $\mathcal{V}$-dcpo D we can take suprema of directed families $\alpha : I \to D$. It is a striking feature of directed complete posets that this act is reflected in the *category* of dcpos, although it does require us to specify an appropriate notion of one dcpo being "below" another one. This notion will be exactly that of an embedding projection pair. The technical results developed in this section will find application in the construction of Scott's D_∞, a model of the untyped λ-calculus, as discussed in Section 5.1.

A priori it is not clear that D_∞ should exist in predicative univalent foundations and it is one of the contributions of this work that this is indeed possible. Our construction largely follows the classical development of Scott's original paper [Sco72], but with some crucial differences. First of all, we carefully keep track of the universe parameters and try to be as general as possible. In the particular case of Scott's D_∞ model of the untyped λ-calculus, we obtain a $\mathcal{U}_0$-dcpo whose carriers lives in the second universe $\mathcal{U}_1$. Secondly, difference arises from proof relevance and these complications are tackled with techniques in univalent foundations and Theorem 2.6.9 in particular, as discussed right before Lemma 3.6.12. Finally, we we generalise Scott's treatment from sequential bilimits to directed bilimits.

Definition 3.6.1 (Deflation). An endofunction $f : D \to D$ on a poset D is a *deflation* if $f(x) \sqsubseteq x$ for all $x : D$.

Definition 3.6.2 (Embedding-projection pair)**.** An *embedding-projection pair* from a $\mathcal{V}$-dcpo D to a $\mathcal{V}$-dcpo E consists of two Scott continuous functions $\varepsilon : D \to E$ (the *embedding*) and $\pi : E \to D$ (the *projection*) such that:

(i) ε is a section of π, and

(ii) $\varepsilon \circ \pi$ is a deflation.

For the remainder of this section, fix the following setup, where we try to be as general regarding universe levels as we can be. We fix a directed preorder $(I, \sqsubseteq)$ with $I : \mathcal{V}$ and $\sqsubseteq$ takes values in some universe $\mathcal{W}$. Now suppose that $(I, \sqsubseteq)$ indexes a family of $\mathcal{V}$-dcpos with embedding-projection pairs between them, i.e. we have

- for every $i : I$, a $\mathcal{V}$-dcpo $D_i : \mathcal{V}\text{-DCPO}_{\mathcal{U},\mathcal{T}}$, and
- for every $i, j : I$ with $i \sqsubseteq j$, an embedding-projection pair $(\varepsilon_{i,j}, \pi_{i,j})$ from D_i to D_j.

Moreover, we require that the following compatibility conditions hold:

$$\text{for every } i : I, \text{ we have } \varepsilon_{i,i} = \pi_{i,i} = \mathrm{id}; \tag{3.6.3}$$

$$\text{for every } i \sqsubseteq j \sqsubseteq k \text{ in } I, \text{ we have } \varepsilon_{i,k} \sim \varepsilon_{j,k} \circ \varepsilon_{i,j} \text{ and } \pi_{i,k} \sim \pi_{i,j} \circ \pi_{j,k}. \tag{3.6.4}$$

Example 3.6.5. If $I :\equiv \mathbb{N}$ with the usual ordering, then we are looking at a diagram of $\mathcal{V}$-dcpos like this

$$\begin{tikzcd} D_0 \arrow[r, hook, shift left, "\varepsilon_{0,1}"] & D_1 \arrow[l, two heads, shift left, "\pi_{0,1}"] \arrow[r, hook, shift left, "\varepsilon_{1,2}"] & D_2 \arrow[l, two heads, shift left, "\pi_{1,2}"] \arrow[r, hook, shift left, "\varepsilon_{2,3}"] & D_3 \arrow[l, two heads, shift left, "\pi_{2,3}"] \arrow[r, hook, shift left, "\varepsilon_{3,4}"] & \cdots \arrow[l, two heads, shift left, "\pi_{3,4}"] \end{tikzcd}$$

where, for example, we have not pictured $\varepsilon_{1,1} : D_1 \hookrightarrow D_1$ and $\varepsilon_{0,2} : D_0 \hookrightarrow D_2$ explicitly, as they are equal to $\mathrm{id}_{D_1} : D_1 \to D_1$ and the composition of $D_0 \xhookrightarrow{\varepsilon_{0,1}} D_1$ and $D_1 \xhookrightarrow{\varepsilon_{1,2}} D_2$, respectively.

The goal is now to construct another $\mathcal{V}$-dcpo D_∞ with embedding-projections pairs $(\varepsilon_{i,\infty} : D_1 \hookrightarrow D_\infty, \pi_{i,\infty} : D_\infty \to D_i)$ for every $i : I$, such that $(D_\infty, (\varepsilon_{i,\infty})_{i:I})$ is the colimit of the diagram given by $(\varepsilon_{i,j})_{i \sqsubseteq j \text{ in } I}$ and $(D_\infty, (\pi_{i,\infty})_{i:I})$ is the limit of the diagram given by $(\pi_{i,j})_{i \sqsubseteq j \text{ in } I}$. In other words, $(D_\infty, (\varepsilon_{i,\infty})_{i:I}, (\pi_{i,\infty})_{i:I})$ is both the colimit and the limit in the category of $\mathcal{V}$-dcpos with embedding-projections pairs between them. We say that it is the *bilimit*.

Definition 3.6.6 (D_∞)**.** We define a poset D_∞ as follows. Its carrier is given by dependent functions $\sigma : \Pi_{i:I} D_i$ satisfying $\pi_{i,j}(\sigma_j) = \sigma_i$ whenever $i \sqsubseteq j$. That is, the carrier is the type

$$\sum_{\sigma : \Pi_{i:I} D_i} \prod_{i,j:I, i \sqsubseteq j} \pi_{i,j}(\sigma_j) = \sigma_i.$$

Note that this defines a subtype of $\Pi_{i:I} D_i$ as the condition $\prod_{i,j:I,i\sqsubseteq j} \pi_{i,j}(\sigma_j) = \sigma_i$ is a property by Theorem 2.3.17 and the fact that each D_i is a set.

These functions are ordered pointwise, i.e. if $\sigma, \tau : \Pi_{i:I} D_i$, then $\sigma \sqsubseteq_{D_\infty} \tau$ exactly when $\sigma_i \sqsubseteq_{D_i} \tau_i$ for every $i : I$.

Lemma 3.6.7. *The poset D_∞ is $\mathcal{V}$-directed complete with suprema calculated pointwise. Paying attention to the universe levels involved, we have $D_\infty : \mathcal{V}\text{-DCPO}_{\mathcal{U}\sqcup\mathcal{V}\sqcup\mathcal{W},\mathcal{U}\sqcup\mathcal{T}}$.*

Proof. If $\alpha : A \to D_\infty$ is a directed family, then the family $\alpha_i : A \to D_i$ given by $\alpha_i(a) :\equiv (\alpha(a))_i$ is directed again, and we define the supremum of α in D_∞ as the function $i \mapsto \bigsqcup \alpha_i$. To see that this indeed defines an element of D_∞, observe that for every $i, j : I$ with $i \sqsubseteq j$ we have

$$\begin{aligned}
\pi_{i,j}\big((\textstyle\bigsqcup \alpha)_j\big) &\equiv \pi_{i,j}\big(\textstyle\bigsqcup \alpha_j\big) \\
&= \textstyle\bigsqcup \pi_{i,j} \circ \alpha_j && \text{(by Scott continuity of } \pi_{i,j}) \\
&\equiv \textstyle\bigsqcup_{a:A} \big(\pi_{i,j}\big((\alpha(a))_j\big)\big) \\
&= \textstyle\bigsqcup \alpha_i && \text{(as } \alpha(a) \text{ is an element of } D_\infty),
\end{aligned}$$

as desired. □

Remark 3.6.8. We allow for general universe levels here, which is why D_∞ lives in the relatively complicated universe $\mathcal{U} \sqcup \mathcal{V} \sqcup \mathcal{W}$. In concrete examples, the situation often simplifies. E.g., in Section 5.1 we find ourselves in the favourable situation described in Remark 3.5.7 where $\mathcal{V} \equiv \mathcal{W} \equiv \mathcal{U}_0$ and $\mathcal{U} \equiv \mathcal{T} \equiv \mathcal{U}_1$, so that we get $D_\infty : \mathcal{U}_0\text{-DCPO}_{\mathcal{U}_1,\mathcal{U}_1}$, as the bilimit of a diagram of dcpos $D_n : \mathcal{U}_0\text{-DCPO}_{\mathcal{U}_1,\mathcal{U}_1}$ indexed by natural numbers.

Definition 3.6.9 ($\pi_{i,\infty}$)**.** For every $i : I$, we define the Scott continuous function $\pi_{i,\infty} : D_\infty \to D_i$ by $\sigma \mapsto \sigma_i$.

Lemma 3.6.10. *The map $\pi_{i,\infty} : D_\infty \to D_i$ is Scott continuous for every $i : I$.*

Proof. This holds because suprema in D_∞ are calculated pointwise and $\pi_{i,\infty}$ selects the i-th component. □

While we could closely follow [Sco72] up until this point, we will now need a new idea to proceed. Our goal is to define maps $\varepsilon_{i,\infty} : D_i \to D_\infty$ for every $i : I$ so that $\varepsilon_{i,\infty}$ and $\pi_{i,\infty}$ form an embedding-projection pair. We give an outline of the idea for defining this map $\varepsilon_{i,\infty}$. For an arbitrary element $x : D_i$, we need to construct $\sigma : D_\infty$ at component $j : I$, say. If we had $k : I$ such that $i, j \sqsubseteq k$, then we could define $\sigma_j : D_j$ by $\pi_{j,k}\big(\varepsilon_{i,k}(x)\big)$. Now semidirectedness of I tells us that there exists such a $k : I$, so the point is to somehow make use of this propositionally truncated fact. This is where Theorem 2.6.9 comes in. We define a map $\kappa^x_{i,j} : (\Sigma_{k:I}\, (i \sqsubseteq k) \times (j \sqsubseteq k)) \to D_j$ by sending k to $\pi_{j,k}\big(\varepsilon_{i,k}(x)\big)$ and show it to be constant, so that it factors through the truncation of its domain. In the special case that $I \equiv \mathbb{N}$, as in [Sco72], we could simply take k to be the sum of the natural numbers i and j, but this does not work in the more general directed case, of course.

Definition 3.6.11 ($\kappa_{i,j}^x$)**.** For every $i, j : I$ and $x : D_i$ we define the function

$$\kappa_{i,j}^x : (\Sigma_{k:I}\,(i \sqsubseteq k) \times (j \sqsubseteq k)) \to D_j$$

by mapping $k : I$ with $i, j \sqsubseteq k$ to $\pi_{j,k}\big(\varepsilon_{i,k}(x)\big)$.

Lemma 3.6.12. *The function $\kappa_{i,j}^x$ is constant for ever $i, j : I$ and $x : D_i$. Hence, $\kappa_{i,j}^x$ factors through $\exists_{k:I}\,(i \sqsubseteq k) \times (j \sqsubseteq k)$ by Theorem 2.6.9.*

Proof. If we have $k_1, k_2 : I$ with $i \sqsubseteq k_1, k_2$ and $j \sqsubseteq k_1, k_2$, then by semidirectedness of I, there exists some $k : K$ with $k_1, k_2 \sqsubseteq k$ and hence,

$$\begin{aligned}
\big(\pi_{j,k_1} \circ \varepsilon_{i,k_1}\big)(x) &= \big(\pi_{j,k_1} \circ \pi_{k_1,k} \circ \varepsilon_{k_1,k} \circ \varepsilon_{i,k_1}\big)(x) && \text{(since } \varepsilon_{k_1,k} \text{ is a section of } \pi_{k_1,k}\text{)}\\
&= \big(\pi_{j,k} \circ \varepsilon_{i,k}\big)(x) && \text{(by Equation 3.6.4)}\\
&= \big(\pi_{j,k} \circ \pi_{k_2,k} \circ \varepsilon_{k_2,k} \circ \varepsilon_{i,k_2}\big)(x) && \text{(since } \varepsilon_{k_2,k} \text{ is a section of } \pi_{k_2,k}\text{)}\\
&= \big(\pi_{j,k_2} \circ \varepsilon_{i,k_2}\big)(x) && \text{(by Equation 3.6.4),}
\end{aligned}$$

proving that $\kappa_{i,j}^x$ is constant. □

Definition 3.6.13 ($\rho_{i,j}$)**.** For every $i, j : I$, the type $\exists_{k:I}\,(i \sqsubseteq k) \times (j \sqsubseteq k)$ has an element since $(I, \sqsubseteq)$ is directed. Thus, Lemma 3.6.12 tells us that we have a function $\rho_{i,j} : D_i \to D_j$ such that if $i, j \sqsubseteq k$, then the equation

$$\rho_{i,j}(x) = \kappa_{i,j}^x(k) \equiv \pi_{j,k}\big(\varepsilon_{i,k}(x)\big) \tag{3.6.14}$$

holds for every $x : D_i$.

Definition 3.6.15 ($\varepsilon_{i,\infty}$)**.** The map ρ induces a map $\varepsilon_{i,\infty} : D_i \to D_\infty$ by sending $x : D_i$ to the function $\lambda\, j : I\,.\,\rho_{i,j}(x)$. To see that this is well-defined, assume that we have $j_1 \sqsubseteq j_2$ in J and $x : D_i$. We have to show that $\pi_{j_1,j_2}\Big(\big(\varepsilon_{i,\infty}(x)\big)_{j_2}\Big) = \big(\varepsilon_{i,\infty}(x)\big)_{j_1}$. By semidirectedness of I and the fact that are looking to prove a proposition, we may assume to have $k : I$ with $i \sqsubseteq k$ and $j_1 \sqsubseteq j_2 \sqsubseteq k$. Then,

$$\begin{aligned}
\pi_{j_1,j_2}\Big(\big(\varepsilon_{i,\infty}(x)\big)_{j_2}\Big) &\equiv \pi_{j_1,j_2}\big(\rho_{i,j_2}(x)\big)\\
&= \pi_{j_1,j_2}\big(\pi_{j_2,k}\big(\varepsilon_{i,k}(x)\big)\big) && \text{(by Equation 3.6.14)}\\
&= \pi_{j_1,k}\big(\varepsilon_{i,k}(x)\big) && \text{(by Equation 3.6.4)}\\
&= \rho_{i,j_1}(x) && \text{(by Equation 3.6.14)}\\
&\equiv \big(\varepsilon_{i,\infty}(x)\big)_{j_1}
\end{aligned}$$

as desired.

This completes the definition of $\varepsilon_{i,\infty}$. From this point on, we can typically work with it by using Equation 3.6.14 and the fact that $\big(\varepsilon_{i,\infty}(x)\big)_j$ is defined as $\rho_{i,j}(x)$.

Lemma 3.6.16. *The map $\rho_{i,j} : D_i \to D_j$ is Scott continuous for every $i, j : I$.*

Proof. Since we are proving a property, we may use semidirectedness of I to get $k : I$ with $i, j \sqsubseteq k$. Then, $\rho_{i,j} \sim \pi_{j,k} \circ \varepsilon_{i,k}$ by Equation 3.6.14. But the functions $\pi_{j,k}$ and $\varepsilon_{i,k}$ are continuous and continuity is preserved by function composition, so $\rho_{i,j}$ is continuous, as we wished to show. □

Lemma 3.6.17. *The map $\varepsilon_{i,\infty} : D_i \to D_\infty$ is Scott continuous for every $i : I$.*

Proof. If $\alpha : A \to D_i$ is directed, then for every $j : I$ we have

$$\begin{aligned}
\left(\varepsilon_{i,\infty}(\textstyle\bigsqcup \alpha)\right)_j &\equiv \rho_{i,j}(\textstyle\bigsqcup \alpha) \\
&= \textstyle\bigsqcup \rho_{i,j} \circ \alpha && \text{(by Lemma 3.6.16)} \\
&\equiv \textstyle\bigsqcup_{a:A} \left(\varepsilon_{i,\infty}(\alpha(a))\right)_j \\
&\equiv \left(\textstyle\bigsqcup (\varepsilon_{i,\infty} \circ \alpha)\right)_j && \text{(as suprema in } D_\infty \text{ are calculated pointwise).}
\end{aligned}$$

Hence, $\varepsilon_{i,\infty}(\bigsqcup \alpha) = \bigsqcup(\varepsilon_{i,\infty} \circ \alpha)$ and $\varepsilon_{i,\infty}$ is seen to be Scott continuous. □

Theorem 3.6.18. *For every $i : I$, the pair $(\varepsilon_{i,\infty}, \pi_{i,\infty})$ is an embedding-projection pair from D_i to D_∞.*

Proof. Scott continuity of both maps is given by Lemmas 3.6.10 and 3.6.17. To see that $\varepsilon_{i,\infty}$ is a section of $\pi_{i,\infty}$, observe that for every $x : D_i$, we have

$$\begin{aligned}
\pi_{i,\infty}\left(\varepsilon_{i,\infty}(x)\right) &\equiv \left(\varepsilon_{i,\infty}(x)\right)_i \\
&\equiv \rho_{i,i}(x) \\
&= \pi_{i,i}\left(\varepsilon_{i,i}(x)\right) && \text{(by Equation 3.6.14)} \\
&\equiv x && \text{(by Equation 3.6.3),}
\end{aligned}$$

so that $\varepsilon_{i,\infty}$ is indeed a section of $\pi_{i,\infty}$. It remains to prove that $\varepsilon_{i,\infty}(\pi_{i,\infty}(\sigma)) \sqsubseteq \sigma$ for every $\sigma : D_\infty$. The order is given pointwise, so let $j : I$ be arbitrary and since we are proving a proposition, assume that we have $k : I$ with $i, j \sqsubseteq k$. Then,

$$\begin{aligned}
\left(\varepsilon_{i,\infty}\left(\pi_{i,\infty}(\sigma)\right)\right)_j &\equiv \left(\varepsilon_{i,\infty}(\sigma_i)\right)_j \\
&\equiv \rho_{i,j}(\sigma_i) \\
&= \pi_{j,k}\left(\varepsilon_{i,k}(\sigma_i)\right) && \text{(by Equation 3.6.14)} \\
&= \pi_{j,k}\left(\varepsilon_{i,k}\left(\pi_{i,k}(\sigma_k)\right)\right) && \text{(since } \sigma \text{ is an element of } D_\infty)
\end{aligned}$$

But $\pi_{i,k} \circ \varepsilon_{i,k}$ is deflationary and $\pi_{j,k}$ is monotone, so

$$\begin{aligned}
&\sqsubseteq \pi_{j,k}(\sigma_k) \\
&= \sigma_j && \text{(since } \sigma \text{ is an element of } D_\infty\text{),}
\end{aligned}$$

finishing the proof. □

Lemma 3.6.19. *The maps $\pi_{i,\infty}$ and $\varepsilon_{i,\infty}$ respectively commute with $\pi_{i,j}$ and $\varepsilon_{i,j}$ whenever $i \sqsubseteq j$, viz. the diagrams*

$$\begin{tikzcd} D_\infty \arrow[rr, "\pi_{i,\infty}"] \arrow[dr, "\pi_{j,\infty}"'] & & D_i \\ & D_j \arrow[ur, "\pi_{i,j}"'] & \end{tikzcd} \qquad \begin{tikzcd} D_i \arrow[rr, "\varepsilon_{i,\infty}"] \arrow[dr, "\varepsilon_{i,j}"'] & & D_\infty \\ & D_j \arrow[ur, "\varepsilon_{j,\infty}"'] & \end{tikzcd}$$

commute for all $i, j : I$ with $i \sqsubseteq j$.

Proof. If $i \sqsubseteq j$ and $\sigma : D_\infty$ is arbitrary, then

$$\pi_{i,j}\left(\pi_{j,\infty}(\sigma)\right) \equiv \pi_{i,j}\left(\sigma_j\right) = \sigma_i$$

precisely because σ is an element of D_∞, which proves the commutativity of the first diagram. For the second, let $x : D_i$ be arbitrary and we compare $\left(\varepsilon_{j,\infty}\left(\varepsilon_{i,j}(x)\right)\right)$ and $\varepsilon_{i,\infty}(x)$ componentwise. So let $j' : I$ be arbitrary. Since we are proving a proposition, we may assume to have $k : I$ with $j, j' \sqsubseteq k$ by semidirectedness of I. We now calculate that

$$\begin{aligned} \left(\varepsilon_{j,\infty}\left(\varepsilon_{i,j}(x)\right)\right)_{j'} &\equiv \rho_{j,j'}\left(\varepsilon_{i,j}(x)\right) \\ &= \pi_{j',k}\left(\varepsilon_{j,k}\left(\varepsilon_{i,j}(x)\right)\right) && \text{(by Equation 3.6.14)} \\ &= \pi_{j',k}\left(\varepsilon_{i,k}(x)\right) && \text{(by Equation 3.6.4)} \\ &= \rho_{i,j'}(x) && \text{(by Equation 3.6.14)} \\ &\equiv \left(\varepsilon_{i,\infty}(x)\right)_{j'} \end{aligned}$$

as desired. □

Theorem 3.6.20. *The $\mathcal{V}$-dcpo D_∞ with the maps $\left(\pi_{i,\infty}\right)_{i:I}$ is the limit of the diagram $\left((D_i)_{i:I}, \left(\pi_{i,j}\right)_{i \sqsubseteq j}\right)$. That is, given a $\mathcal{V}$-dcpo E : $\mathcal{V}$-DCPO$_{\mathcal{U}',\mathcal{T}'}$ and Scott continuous functions $f_i : E \to D_i$ for every $i : I$ such that the diagram*

$$\begin{tikzcd} E \arrow[rr, "f_i"] \arrow[dr, "f_j"'] & & D_i \\ & D_j \arrow[ur, "\pi_{i,j}"'] & \end{tikzcd} \tag{3.6.21}$$

commutes for every $i \sqsubseteq j$, we have a unique Scott continuous function $f_\infty : E \to D_\infty$ making the diagram

$$\begin{tikzcd} E \arrow[rr, "f_i"] \arrow[dr, dashed, "f_\infty"'] & & D_i \\ & D_\infty \arrow[ur, "\pi_{i,\infty}"'] & \end{tikzcd} \tag{3.6.22}$$

commute for every $i : I$.

Proof. Note that Equation 3.6.22 dictates that we must have $(f_\infty(y))_i = f_i(y)$ for every $i : I$. Hence, we define $f_\infty : E \to D_\infty$ as $f_\infty(y) :\equiv \lambda\, i : I\ .\ f_i(y)$, which is Scott continuous because each f_i is and suprema are calculated pointwise in D_∞. To see that f_∞ is well-defined, i.e. that $f_\infty(y)$ is indeed an element of D_∞, observe that for every $i \sqsubseteq j$, the equation $\pi_{i,j}\big((f_\infty(y))_j\big) \equiv \pi_{i,j}(f_j(y)) = f_i(y)$ holds because of Equation 3.6.21. □

It should be noted that in the above universal property E can have its carrier in any universe $\mathcal{U}'$ and its order taking values in any universe $\mathcal{T}'$, even though we required all D_i to have their carriers and orders in two fixed universes $\mathcal{U}$ and $\mathcal{T}$, respectively.

Lemma 3.6.23. *If $i \sqsubseteq j$ in I, then $\varepsilon_{i,\infty}(\sigma_i) \sqsubseteq \varepsilon_{j,\infty}(\sigma_j)$ for every $\sigma : D_\infty$.*

Proof. The order of D_∞ is pointwise, so we compare $\varepsilon_{i,\infty}(\sigma_i)$ and $\varepsilon_{j,\infty}(\sigma_j)$ at an arbitrary component $k : I$. We may assume to have $m : I$ such that $j, k \sqsubseteq m$ by semidirectedness of I. We then calculate that

$$
\begin{aligned}
\big(\varepsilon_{i,\infty}(\sigma_i)\big)_k &\equiv \rho_{i,k}(\sigma_i) \\
&= \big(\pi_{k,m} \circ \varepsilon_{i,m}\big)(\sigma_i) && \text{(by Equation 3.6.14)} \\
&= \big(\pi_{k,m} \circ \varepsilon_{i,m} \circ \pi_{i,j}\big)(\sigma_j) && \text{(since } \sigma \text{ is an element of } D_\infty\text{)} \\
&= \big(\pi_{k,m} \circ \varepsilon_{j,m} \circ \varepsilon_{i,j} \circ \pi_{i,j}\big)(\sigma_j) && \text{(by Equation 3.6.4)}
\end{aligned}
$$

But $\varepsilon_{i,j} \circ \pi_{i,j}$ is deflationary and $\pi_{k,m} \circ \varepsilon_{j,m}$ is monotone, so

$$
\begin{aligned}
&\sqsubseteq \big(\pi_{k,m} \circ \varepsilon_{j,m}\big)(\sigma_j) \\
&= \rho_{j,k}(\sigma_j) && \text{(by Equation 3.6.14)} \\
&\equiv \big(\varepsilon_{j,\infty}(\sigma_j)\big)_k,
\end{aligned}
$$

as we wished to show. □

Lemma 3.6.24. *Every element $\sigma : D_\infty$ is equal to the directed supremum $\bigsqcup_{i:I} \varepsilon_{i,\infty}(\sigma_i)$.*

Proof. The domain of the family is inhabited, because $(I, \sqsubseteq)$ is assumed to be directed. Moreover, if we have $i, j : I$, then there exists $k : I$ with $i, j \sqsubseteq k$, which implies $\varepsilon_{i,\infty}(\sigma_i), \varepsilon_{j,\infty}(\sigma_j) \sqsubseteq \varepsilon_{k,\infty}(\sigma_k)$ by Lemma 3.6.23. Thus, the family $i \mapsto \varepsilon_{i,\infty}(\sigma_i)$ is indeed directed. To see that its supremum is indeed σ we use antisymmetry at an arbitrary component $j : I$. Firstly, observe that

$$
\begin{aligned}
\sigma_j &= \pi_{j,j}\big(\varepsilon_{j,j}(\sigma_j)\big) && \text{(by Equation 3.6.3)} \\
&= \rho_{j,j}(\sigma_j) && \text{(by Equation 3.6.14)} \\
&\equiv \big(\varepsilon_{j,\infty}(\sigma_j)\big)_j \\
&\sqsubseteq \Big(\textstyle\bigsqcup_{i:I} \varepsilon_{i,\infty}(\sigma_i)\Big)_j && \text{(since suprema are computed pointwise in } D_\infty\text{).}
\end{aligned}
$$

Secondly, to prove that $\big(\bigsqcup_{i:I} \varepsilon_{i,\infty}(\sigma_i)\big)_j \sqsubseteq \sigma_j$ it suffices to show that $\big(\varepsilon_{i,\infty}(\sigma_i)\big)_j \sqsubseteq \sigma_j$ for every $i : I$. But this just says that $\varepsilon_{i,\infty} \circ \pi_{i,\infty}$ is a deflation, which was proved in Theorem 3.6.18. □

Although the composites $\varepsilon_{i,\infty} \circ \pi_{i,\infty}$ are deflations for each $i : I$, the supremum of all of them is the identity. This fact will come in useful in Section 5.1.

Lemma 3.6.25. *The family $i \mapsto \varepsilon_{i,\infty} \circ \pi_{i,\infty}$ is directed in the exponential $D_\infty^{D_\infty}$ and its supremum is the identity on D_∞.*

Proof. The order and suprema are given pointwise in exponentials, so this follows from Lemma 3.6.24. ☐

Theorem 3.6.26. *The $\mathcal{V}$-dcpo D_∞ with the maps $(\varepsilon_{i,\infty})_{i:I}$ is the colimit of the diagram $\left((D_i)_{i:I}, (\varepsilon_{i,j})_{i \sqsubseteq j}\right)$. That is, given a $\mathcal{V}$-dcpo $E : \mathcal{V}\text{-DCPO}_{\mathcal{U}',\mathcal{T}'}$ and Scott continuous functions $g_i : D_i \to E$ for every $i : I$ such that the diagram*

$$\begin{tikzcd}
D_i \arrow[rr, "g_i"] \arrow[dr, "\varepsilon_{i,j}"'] & & E \\
& D_j \arrow[ur, "g_j"'] &
\end{tikzcd} \tag{3.6.27}$$

commutes for every $i \sqsubseteq j$, we have a unique Scott continuous function $g_\infty : D_\infty \to E$ making the diagram

$$\begin{tikzcd}
D_i \arrow[rr, "g_i"] \arrow[dr, "\varepsilon_{i,\infty}"'] & & E \\
& D_\infty \arrow[ur, dashed, "g_\infty"'] &
\end{tikzcd} \tag{3.6.28}$$

commute for every $i : I$.

Proof. Note that any such Scott continuous function g_∞ must satisfy

$$\begin{aligned}
g_\infty(\sigma) &= g_\infty\left(\textstyle\bigsqcup_{i:I} \varepsilon_{i,\infty}(\sigma_i)\right) && \text{(by Lemma 3.6.24)} \\
&= \textstyle\bigsqcup_{i:I} g_\infty\left(\varepsilon_{i,\infty}(\sigma_I)\right) && \text{(as } g_\infty \text{ is assumed to be Scott continuous)} \\
&= \textstyle\bigsqcup_{i:I} g_i(\sigma_i) && \text{(by Equation 3.6.28)}
\end{aligned}$$

for every $\sigma : D_\infty$. Accordingly, we define g_∞ by $g_\infty(\sigma) :\equiv \bigsqcup_{i:I} g_i(\sigma_i)$, where we verify that the family is indeed directed: If we have $i, j : I$, then there exists $k : I$ with $i, j \sqsubseteq k$, and we have

$$\begin{aligned}
g_i(\sigma_i) &= g_i\left(\pi_{i,k}(\sigma_k)\right) && \text{(since } \sigma \text{ is an element of } D_\infty) \\
&= g_k\left(\varepsilon_{i,k}\left(\pi_{i,k}(\sigma_k)\right)\right) && \text{(by Equation 3.6.27)} \\
&\sqsubseteq g_k(\sigma_k) && \text{(since } \varepsilon_{i,k} \circ \pi_{i,k} \text{ is deflationary and } g_k \text{ is monotone),}
\end{aligned}$$

and similarly, $g_j(\sigma_j) \sqsubseteq g_k(\sigma_k)$. To see that g_∞ satisfies Equation 3.6.28, let $x : D_i$ be arbitrary and first observe that

$$g_\infty\left(\varepsilon_{i,\infty}(x)\right) \equiv \textstyle\bigsqcup_{j:I} g_j\left(\left(\varepsilon_{i,\infty}(x)\right)_j\right) \equiv \textstyle\bigsqcup_{j:I} g_j\left(\rho_{i,j}(x)\right).$$

We now use antisymmetry to prove that this is equal to $g_i(x)$. In one direction this is easy as $g_i(x) = (g_i \circ \pi_{i,i} \circ \varepsilon_{i,i})(x) \equiv g_i(\rho_{i,i}(x)) \sqsubseteq \bigsqcup_{j:I} g_j(\rho_{i,j}(x))$. In the other direction, it suffices to prove that $g_j(\rho_{i,j}(x)) \sqsubseteq g_i(x)$ for every $j : I$. By directedness of I there exists $k : I$ with $i, j \sqsubseteq k$ so that

$$\begin{aligned} g_j(\rho_{i,j}(x)) &= (g_j \circ \pi_{j,k} \circ \varepsilon_{i,k})(x) && \text{(by Equation 3.6.14)} \\ &= (g_k \circ \varepsilon_{j,k} \circ \pi_{j,k} \circ \varepsilon_{i,k})(x) && \text{(by Equation 3.6.27)} \end{aligned}$$

But $\varepsilon_{j,k} \circ \pi_{j,k}$ is deflationary and g_k is monotone, so

$$\begin{aligned} &\sqsubseteq (g_k \circ \varepsilon_{i,k})(x) \\ &= g_i(x) && \text{(by Equation 3.6.27),} \end{aligned}$$

as we wished to show.

Finally, we verify that g_∞ is Scott continuous. We first check that g_∞ is monotone. If $\sigma \sqsubseteq \tau$ in D_∞, then $g_\infty(\sigma) \equiv \bigsqcup_{i:I} g_i(\sigma_i) \sqsubseteq \bigsqcup_{i:I} g_i(\tau_i) \equiv g_\infty(\tau)$, as each g_i is monotone. It remains to show that $g_\infty(\bigsqcup \alpha) \sqsubseteq \bigsqcup(g_\infty \circ \alpha)$ for every directed family $\alpha : A \to D_\infty$. By definition of g_∞, it suffices to show that $g_i((\bigsqcup \alpha)_i) \sqsubseteq \bigsqcup(g_\infty \circ \alpha)$ for every $i : I$. By continuity of g_i it is enough to establish that $g_i((\alpha(a))_i) \sqsubseteq \bigsqcup(g_\infty \circ \alpha)$ for every $a : A$. But this holds as $g_i((\alpha(a))_i) \sqsubseteq g_\infty(\alpha(a)) \sqsubseteq \bigsqcup(g_\infty \circ \alpha)$, completing our proof. ☐

Proposition 3.6.29. *The bilimit of locally small dcpos is locally small, i.e. if every $\mathcal{V}$-dcpo D_i is locally small for all $i : I$, then so is D_∞.*

Proof. If every D_i is locally small, then for every $i : I$, we have a *specified* $\mathcal{V}$-valued partial order $\sqsubseteq^i_{\mathcal{V}}$ on D_i such that for every $i : I$ and every $x, y : D_i$, we have an equivalence $(x \sqsubseteq_{D_i} y) \simeq (x \sqsubseteq^i_{\mathcal{V}} y)$. Hence, $(\sigma \sqsubseteq_{D_\infty} \tau) \equiv (\Pi_{i:I}(\sigma_i \sqsubseteq_{D_i} \tau_i)) \simeq (\Pi_{i:I}(\sigma_i \sqsubseteq^i_{\mathcal{V}} \tau_i))$, but the latter is small, because $I : \mathcal{V}$ and $\sqsubseteq^i_{\mathcal{V}}$ is $\mathcal{V}$-valued. ☐

3.7 Notes

This chapter is based on our two publications [dJE21a] and [dJon21b], but with an improved exposition and the inclusion of more proofs. More precisely, Sections 3.2 to 3.4 and the exponentials of Section 3.5 feature in both of these works, whereas the material of Section 3.1 and Section 3.6 only appears in [dJE21a], while Theorem 3.5.9 is only included in [dJon21b]. In [dJE21a; dJon21b] the definition of a poset included the requirement that the carrier is a set, because we only realised later that this was redundant (Lemma 3.2.3). Products of dcpos were not discussed in these works, but were, building on our previous work, formalised in AGDA by Brendan Hart [Har20] for a final year MSci project supervised by Martín Escardó and myself.

Sections 3.2, 3.3 and 3.5 are predicative universe-aware adaptations of classical domain theory as expounded in [AJ94; GHK+03], while Section 3.6 is a predicative account of Scott's original paper [Sco72], but with important differences, including an application of Theorem 2.6.9, as discussed at the start of Section 3.6.

Section 3.4 uses the lifting monad in univalent type theory, which originated with the works [EK17; Kna18] that deal with partiality in univalent foundations and aim to avoid (weak) countable choice, which is not provable in constructive univalent foundations [CMR17; Coq18; Swa19b; Swa19a].This is to be contrasted to other approaches to partiality in Martin-Löf Type Theory. The first is Capretta's delay monad [Cap05], which uses coinduction. Arguably, the correct notion of equality of Capretta's delay monad is that of weak bisimilarity where two partial elements are considered equal when they are both undefined, or, when one of them is, so is the other and they have equal values in this case. This prompted the authors of [CUV19] to consider its quotient by weak bisimilarity, but they use countable choice to show that quotient is again a monad. Again using countable choice, they show that their quotient yields free pointed ω-complete posets (ω-cpos). In [ADK17] the authors use a so-called quotient inductive-inductive type (QIIT) to construct the free pointed ω-cpo, essentially by definition of the QIIT. It was shown in [CUV19] that a simpler higher inductive type actually suffices. Regardless, we stress that our approach yields free dcpos as opposed to ω-cpos and does not use countable choice or higher inductive types other than the propositional truncation. But the notion of ω-completeness and countable choice will resurface in our discussion of the Scott model of PCF in Section 5.2.

Chapter 4

Continuous and algebraic dcpos

In the previous chapter we developed sufficient domain theory for the applications considered in Chapter 5, but we have not yet discussed a fundamental topic in domain theory: *algebraic* and *continuous* dcpos. The study of continuous and algebraic dcpo is a rich and deep subject [GHK+03]. We present a treatment of the basic theory and examples in our constructive and predicative approach, where we deal with size issues by taking direct inspiration from category theory and the work of Johnstone and Joyal [JJ82] in particular.

4.1 Introduction

Classically, a dcpo D is said to be *continuous* if for every element x of D the set of elements *way below* it is directed and has supremum x. The problem with this definition in our foundational setup is that the type of elements way below x is not necessarily small. Although this does not stop us from asking it to be directed and having supremum x, this still poses a problem: for example, there would be no guarantee that its supremum is preserved by a Scott continuous function, as it is only required to preserve suprema of directed families indexed by small types.

Our solution is to take inspiration from category theory [JJ82] and to use the ind-completion to give a predicatively suitable definition of continuity of a dcpo. Some care is needed to ensure that the resulting definition expresses a property of a dcpo, rather than an equipment with additional structure. This is of course where the propositional truncation comes in useful, but there are two natural ways of using the truncation. We show that one of them yields a well-behaved notion that serves as our definition of continuity, while the other, which we call *pseudocontinuity*, is problematic in a constructive context. In a classical setting where the axiom of choice is assumed, the two notions (continuity and pseudocontinuity) are equivalent.

Another approach is to turn to the notion of a basis [AJ94, Section 2.2.2], but to include smallness conditions. While we cannot expect the type of elements way below an element x to be small, in many examples it is the case that the type of *basic* elements way below x is small. We show that if a dcpo has a small basis, then it is continuous. In fact, all our running examples of continuous dcpos are actually examples of dcpos with small bases. Moreover, dcpos with small bases are better behaved. For example, they are locally small and so are their exponentials. Furthermore, we show that having a small basis is equivalent to being presented by ideals.

Once we have carefully set up predicatively suitable notions of continuity and small bases, the theory can be developed quite smoothly. Specifically, we discuss

Section 4.2: the way-below relation and compact elements;
Section 4.3: the ind-completion: a tool used in discussing (pseudo) continuity;
Section 4.4: continuity of a dcpo and the interpolation property of the way-below relation;
Section 4.5: pseudocontinuity of a dcpo and issues concerning the axiom of choice;
Section 4.6: algebraicity of a dcpo;
Section 4.7: the notion of a small basis: a strengthening of continuity;
Section 4.8: the notion of a small compact basis: a strengthening of algebraicity;
Section 4.9: examples of dcpos with small compact bases: the type of subsingletons, the lifting of a set, and the powerset; and an example of an algebraic dpco that does not necessarily have a small basis;
Section 4.10: the (rounded) ideal completion of an abstract basis, including an example of a dcpo with a small basis that is not algebraic: the ideal completion of inductively defined dyadic rationals;
Section 4.11: the ideal completion of a small (compact) basis and its relation to the original dcpo;
Section 4.12: bilimits of structurally continuous or algebraic dcpos (with small (compact) bases); and
Section 4.13: exponentials of sup-complete dcpos with small (compact) bases.

4.2 The way-below relation and compactness

The way-below relation is the fundamental ingredient in the development of continuous dcpos. Following Scott [Sco70], we intuitively think of $x \ll y$ as saying that every computation of y has to print x, or something better than x, at some stage.

Definition 4.2.1 (Way-below relation, $x \ll y$)**.** An element x of a $\mathcal{V}$-dcpo D is *way below* an element y of D if whenever we have every directed family $\alpha : I \to D$ indexed by $I : \mathcal{V}$ such that $y \sqsubseteq \bigsqcup \alpha$, then there exists $i : I$ such that $x \sqsubseteq \alpha_i$ already. We denote this situation by $x \ll y$.

Lemma 4.2.2. *The way-below relation enjoys the following properties:*

(i) it is proposition-valued;
(ii) if $x \ll y$, then $x \sqsubseteq y$;
(iii) if $x \sqsubseteq y \ll v \sqsubseteq w$, then $x \ll w$;
(iv) it is antisymmetric;
(v) it is transitive.

Proof. (i) By Theorem 2.3.17 and the fact that we propositionally truncated the existence of $i : I$ in the definition. (ii) Simply take $\alpha : \mathbf{1}_{\mathcal{V}} \to D$ to be $u \mapsto y$. (iii) Suppose that $\alpha : I \to D$ is directed with $w \sqsubseteq \bigsqcup \alpha$. Then $v \sqsubseteq \bigsqcup \alpha$, so by assumption that $y \ll v$ there exists $i : I$ with $y \sqsubseteq \alpha_i$ already. But then $x \sqsubseteq \alpha_i$. (iv) Follows from (ii). (v) Follows from (ii) and (iii). □

In general, the way below relation is not reflexive. The elements for which it is have a special status and are called compact. We illustrate this notion by a series of examples.

Definition 4.2.3 (Compactness)**.** An element is *compact* if it is way below itself.

Example 4.2.4. The least element of a pointed dcpo is always compact.

Example 4.2.5 (Compact elements in $\Omega_{\mathcal{V}}$)**.** The compact elements of $\Omega_{\mathcal{V}}$ are exactly $\mathbf{0}_{\mathcal{V}}$ and $\mathbf{1}_{\mathcal{V}}$. In other words, the compact elements of $\Omega_{\mathcal{V}}$ are precisely the decidable propositions.

Proof. By Example 4.2.4 we know that $\mathbf{0}_{\mathcal{V}}$ must be compact. For $\mathbf{1}_{\mathcal{V}}$, suppose that we have $Q_{(-)} : I \to \Omega_{\mathcal{V}}$ directed such that $\mathbf{1}_{\mathcal{V}} \sqsubseteq \exists_{i:I} Q_i$. Then there exists $i : I$ such that Q_i holds, and hence, $\mathbf{1}_{\mathcal{V}} \sqsubseteq Q_i$. Now suppose that $P : \Omega_{\mathcal{V}}$ is compact. We show that P is decidable. The family $\alpha : (P + \mathbf{1}_{\mathcal{V}}) \to \Omega_{\mathcal{V}}$ given by $\mathrm{inl}(p) \mapsto \mathbf{1}_{\mathcal{V}}$ and $\mathrm{inr}(\star) \mapsto \mathbf{0}_{\mathcal{V}}$ is directed and $P \sqsubseteq \bigsqcup \alpha$. Hence, by compactness, there exists $i : P + \mathbf{1}_{\mathcal{V}}$ such that $P \sqsubseteq \alpha_i$ already. Since being decidable is a property of a proposition, we actually get such an i and by case distinction on it we get decidability of P. □

Example 4.2.6 (Compact elements in the lifting)**.** An element (P, φ) of the lifting $\mathcal{L}_{\mathcal{V}}(X)$ of a set $X : \mathcal{U}$ is compact if and only if P is decidable. Hence, the compact elements of $\mathcal{L}_{\mathcal{V}}(X)$ are exactly $\bot$ and $\eta(x)$ for $x : X$.

Proof. To see that compactness implies decidability of the domain of the partial element, we proceed as in the proof of Example 4.2.5, but for a partial element (P, φ), we consider the family $\alpha : (P + \mathbf{1}_{\mathcal{V}}) \to \mathcal{L}_{\mathcal{V}}(X)$ given by $\mathrm{inl}(p) \mapsto \eta(\varphi(p))$ and $\mathrm{inr}(\star) \mapsto \bot$. Conversely, if we have a partial element (P, φ) with P decidable, then either P is false in which case $(P, \varphi) = \bot$ which is compact by Example 4.2.4, or P holds. So suppose that P holds and let $\alpha : I \to \mathcal{L}_{\mathcal{V}}(X)$ be directed with $P \sqsubseteq \bigsqcup \alpha$. Since P holds, the element $\bigsqcup \alpha$ must be defined, which means that there exists $i : I$ such that α_i is defined. But for this $i : I$ we also have $\bigsqcup \alpha = \alpha_i$ by construction of the supremum, and hence, $P \sqsubseteq \alpha_i$, proving compactness of (P, φ). □

For characterising the compact elements of the powerset, we introduce a lemma, as well as the notion of Kuratowski finiteness and the induction principle for Kuratowski finite subsets.

Lemma 4.2.7. *The compact elements of a dcpo are closed under (existing) binary joins.*

Proof. Suppose that x and y are compact elements of a $\mathcal{V}$-dcpo D, let z be their least upper bound and suppose that we have $\alpha : I \to D$ directed with $z \sqsubseteq \bigsqcup \alpha$. Then $x \sqsubseteq \bigsqcup \alpha$ and $y \sqsubseteq \bigsqcup \alpha$, so by compactness there exist $i : I$ and $j : J$ such that $x \sqsubseteq \alpha_i$ and $y \sqsubseteq \alpha_j$. By semidirectedness of α, there exists $k : I$ with $\alpha_i, \alpha_j \sqsubseteq \alpha_k$, so that $x, y \sqsubseteq \alpha_k$. But z is the join of x and y, so $z \sqsubseteq \alpha_k$, as desired. □

Definition 4.2.8 (Total space of a subset, $\mathbb{T}$)**.** The *total space* of a $\mathcal{T}$-valued subset S of a type X is defined as $\mathbb{T}(S) :\equiv \Sigma_{x:X}(x \in S)$.

Definition 4.2.9 (Kuratowski finiteness)**.**
(i) A type X is *Kuratowski finite* if there exists some natural number $n : \mathbb{N}$ and a surjection $e : \mathrm{Fin}(n) \twoheadrightarrow X$, where $\mathrm{Fin}(n)$ is the inductively defined type with exactly n elements.
(ii) A subset is *Kuratowski finite* if its total space is a Kuratowski finite type.

Thus, a type X is *Kuratowski finite* if its elements can be finitely enumerated, possibly with repetitions, although the repetitions can be removed when X has decidable equality.

Lemma 4.2.10. *The Kuratowski finite subsets of a set are closed under finite unions and contain all singletons.*

Proof. The empty set and any singleton are clearly Kuratowski finite. Moreover, if A and B are Kuratowski finite subsets, then we may assume to have natural numbers n and m and surjections $\sigma : \mathrm{Fin}(n) \twoheadrightarrow \mathbb{T}(A)$ and $\tau : \mathrm{Fin}(m) \twoheadrightarrow \mathbb{T}(B)$. We can then patch these together to obtain a surjection $\mathrm{Fin}(n+m) \twoheadrightarrow \mathbb{T}(A \cup B)$, as desired. □

Lemma 4.2.11 (Induction for Kuratowski finite subsets)**.** *A property of subsets of a type X holds for all Kuratowski finite subsets of X as soon as*
(i) it holds for the empty set,
(ii) it holds for any singleton subset, and
(iii) it holds for $A \cup B$, whenever it holds for A and B.

Proof. Let Q be a such a property and let A be an arbitrary Kuratowski finite subset of X. Since Q is proposition-valued, we may assume to have a natural number n and a surjection $\sigma : \mathrm{Fin}(n) \twoheadrightarrow \mathbb{T}(A)$. Then the subset A must be equal to the finite join of singletons $\{\sigma_0\} \cup \{\sigma_1\} \cup \cdots \cup \{\sigma_{n-1}\}$, which can be shown to satisfy Q by induction on n, and hence, so must A. □

Definition 4.2.12 (β)**.** For a set $X : \mathcal{U}$, we write $\beta : \mathrm{List}(X) \to \mathcal{P}_{\mathcal{U}}(X)$ for the map inductively defined by $[] \mapsto \emptyset$ and $x :: l \mapsto \{x\} \cup \beta(l)$.

Lemma 4.2.13. *A subset $A : X \to \Omega_{\mathcal{U}}$ of a set $X : \mathcal{U}$ is Kuratowski finite if and only if it is in the image of β.*

Proof. The left to right direction follows from Lemma 4.2.10, while the converse follows easily from the induction principle for Kuratowski finite subsets where we use list concatenation in case (iii). ☐

Example 4.2.14 (Compact elements in $\mathcal{P}_{\mathcal{U}}(X)$)**.** The compact elements of $\mathcal{P}_{\mathcal{U}}(X)$ for a set $X : \mathcal{U}$ are exactly the Kuratowski finite subsets of X.

Proof. Suppose first that $A : \mathcal{P}_{\mathcal{U}}(X)$ is a compact element. The family

$$\left(\Sigma_{l:\mathrm{List}(X)}\, \beta(l) \subseteq A\right) \xrightarrow{\beta\circ\mathrm{pr}_1} \mathcal{P}_{\mathcal{U}}(X)$$

is directed, as it contains $\emptyset$ and we can concatenate lists to establish semidirectedness. Moreover, $\left(\Sigma_{l:\mathrm{List}(X)}\, \beta(l) \subseteq A\right)$ lives in $\mathcal{U}$ and we clearly have $A \subseteq \bigsqcup \beta \circ \mathrm{pr}_1$. So by compactness, there exists $l : \mathrm{List}(X)$ with $\beta(l) \subseteq A$ such that $A \subseteq \beta(l)$ already. But this says exactly that A is Kuratowski finite by Lemma 4.2.13.
For the converse we use the induction principle for Kuratowski finite subsets: the empty set is compact by Example 4.2.4, singletons are easily shown to be compact, and binary unions are compact by Lemma 4.2.7. ☐

We end this section by presenting a few lemmas connecting the way-below relation and compactness to Scott continuous sections.

Lemma 4.2.15. *If we have a Scott continuous retract $D \underset{r}{\overset{s}{\rightleftarrows}} E$, then $y \ll s(x)$ implies $r(y) \ll x$ for every $x : D$ and $y : E$.*

Proof. Suppose that $y \ll s(x)$ and that $x \sqsubseteq \bigsqcup \alpha$ for a directed family $\alpha : I \to D$. Then $s(x) \sqsubseteq s(\bigsqcup \alpha) = \bigsqcup s \circ \alpha$ by Scott continuity of s, so there exists $i : I$ such that $y \sqsubseteq s(\alpha_i)$ already. Now monotonicity of r implies $r(y) \sqsubseteq r(s(\alpha_i)) = \alpha_i$ which completes the proof that $r(y) \ll x$. ☐

Lemma 4.2.16. *The embedding in an embedding-projection pair $D \underset{\pi}{\overset{\varepsilon}{\rightleftarrows}} E$ preserves and reflects the way-below relation, i.e. $x \ll y \iff \varepsilon(x) \ll \varepsilon(y)$. In particular, an element x is compact if and only if $\varepsilon(x)$ is.*

Proof. Suppose that $x \ll y$ in D and let $\alpha : I \to E$ be directed with $\varepsilon(y) \sqsubseteq \bigsqcup \alpha$. Then $y = \pi(\varepsilon(y)) \sqsubseteq \bigsqcup \pi \circ \alpha$ by Scott continuity of π. Hence, there exists $i : I$ such that $x \sqsubseteq \pi(\alpha_i)$. But then $\varepsilon(x) \sqsubseteq \varepsilon(\pi(\alpha_i)) \sqsubseteq \alpha_i$ by monotonicity of ε and the fact that $\varepsilon \circ \pi$ is a deflation. This proves that $x \ll y$. Conversely, if $\varepsilon(x) \ll \varepsilon(y)$, then $x = \pi(\varepsilon(x)) \ll y$ by Lemma 4.2.15. ☐

4.3 The ind-completion

The ind-completion will be a useful tool for phrasing and proving results about directed complete *posets* and is itself a directed complete *preorder*, cf. Lemma 4.3.3. It was introduced by Grothendieck and Verdier in [GV72, Section 8] in the context of category theory, but its role in order theory is discussed in [JJ82, Section 1]. We will also use it in the context of order theory, but our treatment will involve a careful consideration of the universes involved, very similar to the original treatment in [GV72].

Definition 4.3.1 ($\mathcal{V}$-ind-completion $\mathcal{V}$-Ind(X), cofinality, $\lesssim$)**.** The *$\mathcal{V}$-ind-completion* $\mathcal{V}$-Ind(X) of a preorder X is the type of directed families in X indexed by types in $\mathcal{V}$, ordered by cofinality. A directed family $\alpha : I \to X$ is *cofinal* in $\beta : J \to X$ if for every $i : I$, there exists $j : J$ such that $\alpha_i \sqsubseteq \beta_j$, and we denote this by $\alpha \lesssim \beta$.

Lemma 4.3.2. *Cofinality defines a preorder on the ind-completion.*

Proof. Straightforward. □

Lemma 4.3.3. *The $\mathcal{V}$-ind-completion $\mathcal{V}$-Ind(X) of a preorder X is $\mathcal{V}$-directed complete.*

Proof. Suppose that we have a directed family $\alpha : I \to \mathcal{V}\text{-Ind}(X)$ with $I : \mathcal{V}$. Then each α_i is a directed family in X indexed by a type $J_i : \mathcal{V}$. We define the family $\hat{\alpha} : (\Sigma_{i:I} J_i) \to X$ by $(i, j) \mapsto \alpha_i(j)$. It is clear that each α_i is cofinal in $\hat{\alpha}$, and that $\hat{\alpha}$ is cofinal in β if every α_i is cofinal in β. So it remains to show that $\hat{\alpha}$ is in fact an element of $\mathcal{V}$-Ind(X), i.e. that it is directed. Because α and each α_i are directed, I and each J_i are inhabited. Hence, so is the domain of $\hat{\alpha}$. Thus, we show that $\hat{\alpha}$ is semidirected. Suppose we have $(i_1, j_1), (i_2, j_2)$ in the domain of $\hat{\alpha}$. By directedness of α, there exists $i : I$ such that α_{i_1} and α_{i_2} are cofinal in α_i. Hence, there exist $j_1', j_2' : J_i$ with $\alpha_{i_1}(j_1) \sqsubseteq \alpha_i(j_1')$ and $\alpha_{i_2}(j_2) \sqsubseteq \alpha_i(j_2')$. Because the family α_i is directed in X, there exists $j : J_i$ such that $\alpha_i(j_1'), \alpha_i(j_2') \sqsubseteq \alpha_i(j)$. Hence, we conclude that $\hat{\alpha}(i_1, j_1) \equiv \alpha_{i_1}(j_1) \sqsubseteq \alpha_i(j_1') \sqsubseteq \alpha_i(j) \equiv \hat{\alpha}(i, j)$, and similarly for (i_2, j_2), which proves semidirectedness of $\hat{\alpha}$. □

Lemma 4.3.4. *Taking directed suprema defines a monotone map from a $\mathcal{V}$-dcpo to its $\mathcal{V}$-ind-completion, denoted by $\bigsqcup : \mathcal{V}\text{-Ind}(D) \to D$.*

Proof. We have to show that $\bigsqcup \alpha \sqsubseteq \bigsqcup \beta$ for directed families α and β such that α is cofinal in β. Note that the inequality holds as soon as $\alpha_i \sqsubseteq \bigsqcup \beta$ for every i in the domain of α. For this, it suffices that for every such i, there exists a j in the domain of β such that $\alpha_i \sqsubseteq \beta_j$. But the latter says exactly that α is cofinal in β. □

Johnstone and Joyal [JJ82] generalise the notion of continuity from posets to categories and do so by phrasing it in terms of $\bigsqcup : \mathcal{V}\text{-Ind}(D) \to D$ having a left adjoint. We follow their approach and build towards this. It turns out to be convenient to use the following two notions, which are in fact equivalent by Lemma 4.3.7.

Definition 4.3.5 (Approximate, left adjunct)**.** For an element x of a dcpo D and a directed family $\alpha : I \to D$, we say that

(i) α *approximates* x if the supremum of α is x and each α_i is way below x, and
(ii) α is *left adjunct* to x if $\alpha \lesssim \beta \iff x \sqsubseteq \bigsqcup \beta$ for every directed family β.

Remark 4.3.6. For a $\mathcal{V}$-dcpo D, a function $L : D \to \mathcal{V}\text{-Ind}(D)$ is a left adjoint to $\bigsqcup : \mathcal{V}\text{-Ind}(D) \to D$ precisely when $L(x)$ is left adjunct to x for every $x : D$.
One may object at this point and argue that a function $L : D \to \mathcal{V}\text{-Ind}(D)$ needs to be monotone in order to truly be a left adjoint. But monotonicity actually follows from the condition that $L(x)$ is left adjunct to x for every $x : D$, as we show in Corollary 4.3.9.

Lemma 4.3.7. *A directed family α approximates an element x if and only if it is left adjunct to it.*

Proof. Suppose first α approximates x. If $\alpha \lesssim \beta$, then $x = \bigsqcup \alpha \sqsubseteq \bigsqcup \beta$, by Lemma 4.3.4. Conversely, if $x \sqsubseteq \bigsqcup \beta$, then α is cofinal in β: for if i is in the domain of α, then $\alpha_i \ll x$, so there exists j such that $\alpha_i \sqsubseteq \beta_j$ already.
In the other direction, suppose that α is left adjunct to x. We show that each α_i is way below x. If β is a directed family with $x \sqsubseteq \bigsqcup \beta$, then α is cofinal in β as α is assumed to be left adjunct to x. Hence, for every i, there exists j with $\alpha_i \sqsubseteq \beta_j$, proving that $\alpha_i \ll x$. Since α is cofinal in itself, we get $x \sqsubseteq \bigsqcup \alpha$ by assumption. For the other inequality, we note that $x \sqsubseteq \bigsqcup \hat{x}$, where $\hat{x} : \mathbf{1} \to D$ is the directed family that maps to x. Hence, as α is left adjunct to x, we must have that α is cofinal in $\hat{x}$, which means that each α_i is below x. Thus, we conclude $\bigsqcup \alpha \sqsubseteq x$ and $x = \bigsqcup \alpha$, as desired. ☐

Proposition 4.3.8. *For a $\mathcal{V}$-dcpo D, a function $L : D \to \mathcal{V}\text{-Ind}(D)$ is a left adjoint to $\bigsqcup : \mathcal{V}\text{-Ind}(D) \to D$ if and only if $L(x)$ approximates x for every $x : D$.*

Proof. Immediate from Lemma 4.3.7 and Remark 4.3.6. ☐

Corollary 4.3.9. *If a function $L : D \to \mathcal{V}\text{-Ind}(D)$ is a left adjoint to the function $\bigsqcup : \mathcal{V}\text{-Ind}(D) \to D$, then it is monotone. In fact, it is also order-reflecting in this case.*

Proof. Suppose that L is a left adjoint to $\bigsqcup$ and that we have elements $x, y : D$. Since L is a left adjoint, we have $L(x) \lesssim L(y) \iff x \sqsubseteq \bigsqcup L(y)$, but $L(y)$ approximates y, so $\bigsqcup L(y) = y$ and hence $L(x) \lesssim L(y) \iff x \sqsubseteq y$, so L preserves and reflects the order. ☐

4.4 Continuous dcpos

We define what it means for a $\mathcal{V}$-dcpo to be (structurally) continuous and prove the interpolation properties for the way-below relation. We postpone giving examples (see Sections 4.9 and 4.10.2) until we have developed the theory further.

Definition 4.4.1 (Structural continuity). A $\mathcal{V}$-dcpo D is *structurally continuous* if for every $x : D$ we have a specified $I : \mathcal{V}$ and directed *approximating family* $\alpha : I \to D$ such that α has supremum x and each element $\alpha(i)$ is way below x.

Remark 4.4.2 (I_x, α_x). Note how structural continuity equips a dcpo with a function assigning an approximating family to every element of the dcpo. If we have such an equipment, we will write $\alpha_x : I_x \to D$ for the approximating family of an element x.

The somewhat verbose definition of structural continuity can be succinctly phrased as follows.

Proposition 4.4.3. *Structural continuity of a $\mathcal{V}$-dcpo D is equivalent to having a specified left adjoint to $\bigsqcup : \mathcal{V}\text{-Ind}(D) \to D$.*

Proof. Immediate from Proposition 4.3.8. $\square$

Remark 4.4.4. It should be noted that structural continuity is *not* a property because an element $x : D$ can have several different but bicofinal approximating families, e.g. if $\alpha : I \to D$ approximates x, then so does $[\alpha, \alpha] : (I + I) \to D$. In other words, the left adjoint to $\bigsqcup : \mathcal{V}\text{-Ind}(D) \to D$ is not unique, although it is unique up to isomorphism. In category theory this is typically sufficient (and often the best one can do). Johnstone and Joyal follow this philosophy in [JJ82], but we want the type of continuous $\mathcal{V}$-dcpos to be a subtype of the $\mathcal{V}$-dcpos. One reason that a property is preferred is that it is considered good mathematical practice to only consider morphisms that preserve imposed structure. In the case of structural continuity, this would imply preservation of the way-below relation, but this excludes many Scott continuous maps, e.g. if the output of a constant map is not compact, then it does not preserve the way-below relation.
One may ask why the univalence axiom cannot be used to identify these isomorphic objects. The point is that the ind-completion $\mathcal{V}\text{-Ind}(D)$ is *not* a univalent category in the sense of [AKS15], because it is a preorder and not a poset. One way to obtain a subtype is to propositionally truncate the notion of structural continuity and this is indeed the approach that we will take. However, another choice that would yield a property is to identify bicofinal elements of $\mathcal{V}\text{-Ind}(D)$ by quotienting. This approach is discussed at length in Section 4.5 and in particular it is explained to be inadequate in a constructive setting.

Definition 4.4.5 (Continuity of a dcpo). A $\mathcal{V}$-dcpo is *continuous* if the propositional truncation of its structural continuity holds.

Thus, a dcpo is continuous if we have an *unspecified* function assigning an approximating family to every element of the dcpo.

Proposition 4.4.6. *Continuity of a $\mathcal{V}$-dcpo D is equivalent to having an unspecified left adjoint to $\bigsqcup : \mathcal{V}\text{-Ind}(D) \to D$.*

Proof. By Proposition 4.4.3 and functoriality of the propositional truncation. □

Lemma 4.4.7. *For elements x and y of a structurally continuous dcpo, the following are equivalent:*

(i) $x \sqsubseteq y$;
(ii) $\alpha_x(i) \sqsubseteq y$ *for every* $i : I_x$;
(iii) $\alpha_x(i) \ll y$ *for every* $i : I_x$.

Proof. Note that (iii) implies (ii) and (ii) implies (i), because if $\alpha_x(i) \sqsubseteq y$ for every $i : I_x$, then $x = \bigsqcup \alpha_x \sqsubseteq y$, as desired. So it remains to prove that (i) implies (iii), but this holds, because $\alpha_x(i) \ll x$ for every $i : I_x$. □

Lemma 4.4.8. *For elements x and y of a structurally continuous dcpo, x is way below y if and only if there exists $i : I_y$ such that $x \sqsubseteq \alpha_y(i)$.*

Proof. The left-to-right implication holds, because α_y is a directed family with supremum y, while the converse holds because $\alpha_y(i) \ll y$ for every $i : I_y$. □

We now prove three interpolation lemmas for structurally continuous dcpos. Because the conclusions of the lemmas are propositions, the results will follow for continuous dcpo immediately.

Lemma 4.4.9 (Nullary interpolation for the way-below relation)**.** *For every $x : D$ of a (structurally) continuous dcpo D, there exists $y : D$ such that $y \ll x$.*

Proof. The approximating family α_x is directed, so there exists $i : I_x$ and hence we can take $y :\equiv \alpha_x(i)$ since $\alpha_x(i) \ll x$. □

Our proof of the following lemma is inspired by [JJ82, Proposition 2.12].

Lemma 4.4.10 (Unary interpolation for the way-below relation)**.** *If $x \ll y$ in a (structurally) continuous dcpo D, then there exists an* interpolant *$d : D$ such that $x \ll d \ll y$.*

Proof. By structural continuity, we can approximate every approximant $\alpha_y(i)$ of y by an approximating family $\beta_i : J_i \to D$. This defines a map $\hat{\beta}$ from I_y to $\mathcal{V}$-Ind(D), the ind-completion of the $\mathcal{V}$-dcpo D, by sending $i : I_y$ to the directed family β_i. We claim that $\hat{\beta}$ is directed in $\mathcal{V}$-Ind(D). Since α_y is directed, I_y is inhabited, so it remains to prove that $\hat{\beta}$ is semidirected. So suppose we have $i_1, i_2 : I_y$. Because α_y is semidirected, there exists $i : I_y$ such that $\alpha_y(i_1), \alpha_y(i_2) \sqsubseteq \alpha_y(i)$. We claim that β_{i_1} and β_{i_2} are cofinal in β_i, which would prove semidirectedness of $\hat{\beta}$. We give the argument for i_1 only as the case for i_2 is completely analogous. For the cofinality, we have to show that for every $j : J_{i_1}$, there exists $j' : J_i$ such that $\beta_{i_1}(j) \sqsubseteq \beta_i(j')$. But this holds because $\beta_{i_1}(j) \ll \bigsqcup \beta_i$ for every such j, as we have $\beta_{i_1}(j) \ll \alpha_y(i_1) \sqsubseteq \alpha_y(i) \sqsubseteq \bigsqcup \beta_i$.
Thus, $\hat{\beta}$ is directed in $\mathcal{V}$-Ind(D) and hence we can calculate its supremum in $\mathcal{V}$-Ind(D)

to obtain the *directed* family $\gamma : (\Sigma_{i:I} J_i) \to D$ given by $(i, j) \mapsto \beta_i(j)$.
We now show that y is below the supremum of γ. By Lemma 4.4.7, it suffices to prove that $\alpha_y(i) \sqsubseteq \bigsqcup \gamma$ for every $i : I_y$, and, in turn, to prove this for an $i : I_y$ it suffices to prove that $\beta_i(j) \sqsubseteq \bigsqcup \gamma$ for every $j : J_i$. But this is immediate from the definition of γ. Thus, $y \sqsubseteq \bigsqcup \gamma$. Because $x \ll y$, there exists $(i, j) : \Sigma_{i:I} J_i$ such that $x \sqsubseteq \gamma(i, j) \equiv \beta_i(j)$. Finally, for our interpolant, we take $d :\equiv \alpha_y(i)$. Then, indeed, $x \ll d \ll y$, because $x \sqsubseteq \beta_i(j) \ll \alpha_y(i) \equiv d$ and $d \equiv \alpha_y(i) \ll y$, completing the proof. □

Lemma 4.4.11 (Binary interpolation for the way-below relation). *If $x \ll z$ and $y \ll z$ in a (structurally) continuous dcpo D, then there exists an interpolant $d : D$ such that $x, y \ll d$ and $d \ll z$.*

The proof is a straightforward application of unary interpolation.

Proof. Using that $x \ll z$ and $y \ll z$, there exist interpolants $d_1, d_2 : D$ such that $x \ll d_1 \ll z$ and $y \ll d_2 \ll z$. Hence, there exist $i_1, i_2 : I_z$ such that $d_1 \sqsubseteq \alpha_z(i_1)$ and $d_2 \sqsubseteq \alpha_z(i_2)$. By semidirectedness of α_z, there then exists $i : I_z$ for which $d_1, d_2 \sqsubseteq \alpha_z(i)$. Our final interpolant is defined as $d :\equiv \alpha_z(i)$, which works because $x \ll d_1 \sqsubseteq d$, $y \ll d_2 \sqsubseteq d$ and $d \equiv \alpha_z(i) \ll z$. □

Both continuity and structural continuity are closed under Scott continuous retracts. Keeping track of universes, it holds in the following generality:

Theorem 4.4.12. *If we have dcpos $D : \mathcal{V}\text{-DCPO}_{\mathcal{U},\mathcal{T}}$ and $E : \mathcal{V}\text{-DCPO}_{\mathcal{U}',\mathcal{T}'}$ such that D is a Scott continuous retract of E, then D is (structurally) continuous if E is.*

Proof. We prove the result for structural continuity, as the other will follow from that and the fact that the propositional truncation is functorial. So suppose that we have a Scott continuous section $s : D \to E$ and retraction $r : E \to D$ and structural continuity of E. We claim that for every $x : D$, the family $r \circ \alpha_{s(x)}$ is approximating for x. Firstly, it is directed, because $\alpha_{s(x)}$ is and r is Scott continuous. Secondly,

$$\begin{aligned}
\bigsqcup r \circ \alpha_{s(x)} &= r\big(\bigsqcup \alpha_{s(x)}\big) && \text{(by Scott continuity of } r) \\
&= r(s(x)) && \text{(as } \alpha_{s(x)} \text{ is the approximating family of } s(x)) \\
&= x && \text{(because } s \text{ is a section of } r),
\end{aligned}$$

so the supremum of $r \circ \alpha_{s(x)}$ is x. Finally, we must prove that $r\big(\alpha_{s(x)}(i)\big) \ll x$ for every $i : I_x$. By Lemma 4.2.15, this is implied by $\alpha_{s(x)}(i) \ll s(x)$, which holds as $\alpha_{s(x)}$ is the approximating family of $s(x)$. □

Proposition 4.4.13. *A (structurally) continuous dcpo is locally small if and only if its way-below relation has small values.*

Proof. By Lemmas 4.4.7 and 4.4.8, we have

$$x \sqsubseteq y \iff \forall_{i:I_x}(\alpha_x(i) \ll y) \quad \text{and} \quad x \ll y \iff \exists_{i:I_y}\big(x \sqsubseteq \alpha_y(i)\big),$$

for every two elements x and y of a structurally continuous dcpo. But the types I_x and I_y are small, finishing the proof. The result also holds for continuous dcpos, because what we are proving is a proposition. □

Proposition 4.4.13 is significant because the definition of the way-below relation for a $\mathcal{V}$-dcpo D quantifies over all families into D indexed by types in $\mathcal{V}$.

4.5 Pseudocontinuity

In light of Proposition 4.4.3, we see that a $\mathcal{V}$-dcpo D can be structurally continuous in more than one way: the map $\bigsqcup : \mathcal{V}\text{-Ind}(D) \to D$ can have two left adjoints L_1, L_2 such that for some $x : D$, the directed families $L_1(x)$ and $L_2(x)$ are bicofinal, yet unequal. In order for the left adjoint to be truly unique (and not just up to isomorphism), the preorder $\mathcal{V}\text{-Ind}(D)$ would have to identify bicofinal families. Of course, we could enforce this identification by passing to the poset reflection $\mathcal{V}\text{-Ind}(D)/{\approx}$ of $\mathcal{V}\text{-Ind}(D)$ and this section studies exactly that.

Another perspective on the situation is the following: The type-theoretic definition of structural continuity of a $\mathcal{V}$-dcpo D is of the following form $\Pi_{x:D}\Sigma_{I:\mathcal{V}}\Sigma_{\alpha:I\to D}\ldots$, while continuity is defined as its propositional truncation $\|\Pi_{x:D}\Sigma_{I:\mathcal{V}}\Sigma_{\alpha:I\to D}\ldots\|$. Yet another way to obtain a property is by putting the propositional truncation on the *inside* instead: $\Pi_{x:D}\|\Sigma_{I:\mathcal{V}}\Sigma_{\alpha:I\to D}\ldots\|$. We study what this amounts to and how it relates to (structural) continuity and the poset reflection. Our results are summarised in Table 4.5.4 below.

Definition 4.5.1 (Pseudocontinuity)**.** A $\mathcal{V}$-dcpo D is *pseudocontinuous* if for every $x : D$ there exists an unspecified directed family that approximates x.

Note that structural continuity $\Rightarrow$ continuity $\Rightarrow$ pseudocontinuity, but reversing the first implication is an instance of global choice, while reversing the second amounts to an instance of the axiom of choice (Equation 2.7.25) that we do not expect to be provable in our constructive setting. We further discuss this point in Remark 4.5.5.

For a $\mathcal{V}$-dcpo D, the map $\bigsqcup : \mathcal{V}\text{-Ind}(D) \to D$ is monotone, so it induces a unique monotone map $\bigsqcup_{\approx} : \mathcal{V}\text{-Ind}(D)/{\approx} \to D$ such that the diagram

$$\begin{tikzcd} \mathcal{V}\text{-Ind}(D)/{\approx} \arrow[rr, "\bigsqcup_{\approx}"] \arrow[dr, "{[-]}"'] & & D \\ & \mathcal{V}\text{-Ind}(D) \arrow[ur, "\bigsqcup"'] & \end{tikzcd} \tag{4.5.2}$$

commutes.

Proposition 4.5.3. *A $\mathcal{V}$-dcpo D is pseudocontinuous if and only if the map of posets $\bigsqcup_{\approx} : \mathcal{V}\text{-Ind}(D)/{\approx} \to D$ has a (specified) left adjoint.*

Observe that the type of left adjoints to $\bigsqcup_{\approx} : \mathcal{V}\text{-Ind}(D)/{\approx} \to D$ is a proposition, precisely because $\mathcal{V}\text{-Ind}(D)/{\approx}$ is a poset, cf. [AKS15, Lemma 5.2].

Proof. Suppose that $\bigsqcup_{\approx} : \mathcal{V}\text{-Ind}(D)/{\approx} \to D$ has a left adjoint L and let $x : D$ be arbitrary. We have to prove that there exists a directed family $\alpha : I \to D$ that approximates x. By surjectivity of the universal map $[-]$, there exists a directed family $\alpha : I \to D$ such that $L(x) = [\alpha]$. Moreover, α approximates x by virtue of Lemma 4.3.7, since for every $\beta : \mathcal{V}\text{-Ind}(D)$, we have

$$\begin{aligned} \alpha \lesssim \beta &\iff L(x) \leq [\beta] && \text{(since } L(x) = [\alpha]\text{)} \\ &\iff x \sqsubseteq \textstyle\bigsqcup_{\approx} [\beta] && \text{(since } L \text{ is a left adjoint to } \textstyle\bigsqcup_{\approx}\text{)} \\ &\iff x \sqsubseteq \textstyle\bigsqcup \beta && \text{(by Equation 4.5.2).} \end{aligned}$$

The converse is more involved and features another application of Theorem 2.6.9, similar to the proof of Proposition 3.4.14. Assume that D is pseudocontinuous. We start by constructing the left adjoint, so let $x : D$ be arbitrary. Writing $\mathcal{A}_x$ for the type of directed families that approximate x, we have an obvious map $\varphi_x : \mathcal{A}_x \to \mathcal{V}\text{-Ind}(D)$ that forgets that the directed family approximates x.
We claim that all elements in the image of φ_x are bicofinal. For if α and β are directed families both approximating x, then for every i in the domain of α we know that $\alpha_i \ll x = \bigsqcup \beta$, so that there exists j with $\alpha_i \sqsubseteq \beta_j$. Hence, passing to the poset reflection, the composite $[-] \circ \varphi_x$ is constant. Thus, by Theorem 2.6.9 we have a (necessarily unique) map ψ_x making the diagram

$$\begin{tikzcd} \mathcal{A}_x \arrow[rr, "{[-]\circ\varphi_x}"] \arrow[dr, "|-|"'] & & \mathcal{V}\text{-Ind}(D)/{\approx} \\ & \|\mathcal{A}_x\| \arrow[ur, dashed, "\psi_x"'] & \end{tikzcd}$$

commute. Since D is assumed to be pseudocontinuous, we have exactly $\|\mathcal{A}_x\|$ for every $x : D$, so together with ψ_x this defines a map $L : D \to \mathcal{V}\text{-Ind}(D)/{\approx}$ by $L(x) :\equiv \psi_x(p)$, where p witnesses pseudocontinuity at x.
Lastly, we prove that L is indeed a left adjoint to $\bigsqcup_{\approx}$. So let $x : D$ be arbitrary. Since we're proving a property, we can use pseudocontinuity at x to specify a directed family α that approximates x. We now have to prove $[\alpha] \leq \beta' \iff x \sqsubseteq \bigsqcup_{\approx} \beta'$ for every $\beta' : \mathcal{V}\text{-Ind}(D)/{\approx}$. This is a proposition, so using quotient induction once more, it suffices to prove $[\alpha] \leq [\beta] \iff x \sqsubseteq \bigsqcup_{\approx}[\beta]$ for every $\beta : \mathcal{V}\text{-Ind}(D)$. Indeed, for such β we have

$$\begin{aligned} [\alpha] \leq [\beta] &\iff \alpha \lesssim \beta \\ &\iff x \sqsubseteq \textstyle\bigsqcup \beta && \text{(by Lemma 4.3.7 and the fact that } \alpha \text{ approximates } x\text{)} \\ &\iff x \sqsubseteq \textstyle\bigsqcup_{\approx} [\beta] && \text{(by Equation 4.5.2),} \end{aligned}$$

finishing the proof. □

Thus, the explicit type-theoretic formulation and the formulation using left adjoints in each row of Table 4.5.4 (which summarises our findings) are equivalent.

	Type-theoretic formulation	Formulation with adjoints	Property
Struc. cont.	$\Pi_{x:D}\Sigma_{I:\mathcal{V}}\Sigma_{\alpha:I\to D}\,\delta(\alpha,x)$	Specified left adjoint to $\bigsqcup : \mathcal{V}\text{-Ind}(D) \to D$	$\times$
Cont.	$\|\Pi_{x:D}\Sigma_{I:\mathcal{V}}\Sigma_{\alpha:I\to D}\,\delta(\alpha,x)\|$	Unspecified left adjoint to $\bigsqcup : \mathcal{V}\text{-Ind}(D) \to D$	✓
Pseudocont.	$\Pi_{x:D}\|\Sigma_{I:\mathcal{V}}\Sigma_{\alpha:I\to D}\,\delta(\alpha,x)\|$	Specified left adjoint to $\bigsqcup_{\approx} : \mathcal{V}\text{-Ind}(D)/{\approx} \to D$	✓

Table 4.5.4: (Structural) continuity and pseudocontinuity of a dcpo D, where $\delta(\alpha, x)$ abbreviates that α is directed and approximates x.

Remark 4.5.5. The issue with pseudocontinuity is that taking the quotient by bicofinality introduces a dependence on instances of the axiom of choice when it comes to proving properties of pseudocontinuous dcpos. An illustrative example is the proof of unary interpolation (Lemma 4.4.10), where we used structural continuity to first approximate an element y by α_y and then, in turn, approximate every approximant $\alpha_y(i)$. With pseudocontinuity this argument would require *choosing* an approximating family for every i. Another example is that while the preorder $\mathcal{V}$-Ind(D) is $\mathcal{V}$-directed complete, a direct lifting of the proof of this fact to the poset reflection $\mathcal{V}$-Ind$(D)/{\approx}$ requires the axiom of choice. Hence, the Rezk completion [AKS15], of which the poset reflection is a special case, does not necessarily preserve (filtered) colimits. Related issues concerning the axiom of choice are also discussed in [JJ82, pp. 260–261].

4.6 Algebraic dcpos

Many of our examples of dcpos are not just continuous, but satisfy the stronger condition of being algebraic, meaning their elements can be approximated by compact elements only.

Definition 4.6.1 (Structural algebraicity)**.** A $\mathcal{V}$-dcpo D is *structurally algebraic* if for every $x : D$ we have a specified $I : \mathcal{V}$ and directed *compact family* $\kappa : I \to D$ such that κ has supremum x and each element $\kappa(i)$ is compact.

Remark 4.6.2 (I_x, κ_x). Note how structural algebraicity equips a dcpo with a function assigning a compact family to every element of the dcpo. If we have such an equipment, we will write $\kappa_x : I_x \to D$ for the compact family of an element x.

Definition 4.6.3 (Algebraicity)**.** A $\mathcal{V}$-dcpo is *algebraic* if the propositional truncation of its structural algebraicity holds.

Thus, a dcpo is continuous if we have an *unspecified* function assigning a compact family to every element of the dcpo.

Lemma 4.6.4. *Every (structurally) algebraic dcpo is (structurally) continuous.*

Proof. We prove that structurally algebraic dcpos are structurally continuous. The claim for algebraic and continuous then follows by functoriality of the propositional truncation. It suffices to prove that $\kappa_x(i) \ll x$ for every $i : I_x$. By assumption, $\kappa_x(i)$ is compact and has supremum x. Hence, $\kappa_x(i) \ll \kappa_x(i) \sqsubseteq \bigsqcup \kappa_x = x$, so $\kappa_x(i) \ll x$. □

4.7 Small bases

Recall that the traditional, set-theoretic definition of a dcpo D being continuous says that for every element $x \in D$, the subset $\{y \in D \mid y \ll x\}$ is directed with supremum x. As explained in the Introduction of this chapter, the problem with this definition in a predicative context is that the subset $\{y \in D \mid y \ll x\}$ is not small. But, as is well-known in domain theory, it is sufficient (and in fact equivalent) to instead ask that D has a subset B, known as a *basis*, such that the subset $\{b \in B \mid b \ll x\} \subseteq B$ is directed with supremum x, see [AJ94, Section 2.2.2] and [GHK+03, Definition III-4.1]. The idea developed in this section is that in many examples we can find a *small* basis giving us a predicative handle on the situation.

If a dcpo has a small basis, then it is continuous. In fact, all our running examples of continuous dcpos are actually examples of dcpos with small bases. Moreover, dcpos with small bases are better behaved. For example, they are locally small and so are their exponentials, which also have small bases (Section 4.13). Moreover, in Section 4.11 we show that having a small basis is equivalent to being presented by ideals.

Definition 4.7.1 (Small basis)**.** For a $\mathcal{V}$-dcpo D, a map $\beta : B \to D$ with $B : \mathcal{V}$ is a *small basis* for D if the following conditions hold:

(i) for every $x : D$, the family $(\Sigma_{b:B}(\beta(b) \ll x)) \xrightarrow{\beta \circ \mathrm{pr}_1} D$ is directed and has supremum x;
(ii) for every $x : D$ and $b : B$, the proposition $\beta(b) \ll x$ is $\mathcal{V}$-small.

We will write $\downarrow_\beta x$ for the type $\Sigma_{b:B}(\beta(b) \ll x)$ and conflate this type with the canonical map $\downarrow_\beta x \xrightarrow{\beta \circ \mathrm{pr}_1} D$.

Item (ii) ensures not only that the type $\Sigma_{b:B}(\beta(b) \ll x)$ is $\mathcal{V}$-small, but also that a dcpo with a small basis is locally small (Proposition 4.7.5).

Remark 4.7.2. If $\beta : B \to D$ is a small basis for a $\mathcal{V}$-dcpo D, then the type $\downarrow_\beta x$ is small. Hence, we have a type $I : \mathcal{V}$ and an equivalence $\varphi : I \simeq \downarrow_\beta x$ and we see that the family $I \xrightarrow{\varphi} \downarrow_\beta x \xrightarrow{\beta \circ \mathrm{pr}_1} D$ is directed and has the same supremum as $\downarrow_\beta x \to D$. We will use this tacitly and write as if the type $\downarrow_\beta x$ is actually a type in $\mathcal{V}$.

Lemma 4.7.3. *If a dcpo comes equipped with a small basis, then it is structurally continuous. Hence, if a dpco has an unspecified small basis, then it is continuous.*

Proof. For every element x of a dcpo D, the family $\downarrow_\beta x \to D$ approximates x, so the assignment $x \mapsto \downarrow_\beta x$ makes D structurally continuous. $\square$

Lemma 4.7.4. *In a dcpo D with a small basis $\beta : B \to D$, we have $x \sqsubseteq y$ if and only if $\forall_{b:B}(\beta(b) \ll x \to \beta(b) \ll y)$ for every $x, y : D$.*

Proof. If $x \sqsubseteq y$ and $\beta(b) \ll x$, then $\beta(b) \ll y$, so the left-to-right implication is clear. For the converse, suppose that the condition of the lemma holds. Because $x = \bigsqcup \downarrow_\beta x$, the inequality $x \sqsubseteq y$ holds as soon as $\beta(b) \sqsubseteq y$ for every $b : B$ with $\beta(b) \ll x$, but this is implied by the condition. $\square$

Proposition 4.7.5. *A dcpo with a small basis is locally small. Moreover, the way-below relation on all of the dcpo has small values.*

Proof. The first claim follows from Lemma 4.7.4 and the second follows from the first and Proposition 4.4.13. $\square$

A notable feature of dcpos with a small basis is that interpolants for the way-below relation, cf. Lemmas 4.4.9 to 4.4.11, can be found in the basis, as we show now.

Lemma 4.7.6 (Nullary interpolation in the basis for the way-below relation)**.** *In a dcpo D with a small basis $\beta : B \to D$, there exists $b : B$ with $\beta(b) \ll x$ for every $x : D$.*

Proof. For every $x : D$, the approximating family $\downarrow x$ is directed, so there exists $b : B$ with $\beta(b) \ll x$. $\square$

Lemma 4.7.7 (Unary interpolation in the basis for the way-below relation)**.** *If $x \ll y$ in a dcpo D with a small basis $\beta : B \to D$, then there exists an interpolant $b : B$ such that $x \ll \beta(b) \ll y$.*

Proof. The small basis ensures that D is structurally continuous by Lemma 4.7.3. Hence, if $x \ll y$, then there exists an interpolant $d : D$ with $x \ll d \ll y$. Now $d \ll y \sqsubseteq \bigsqcup \downarrow_\beta y$, so there exists $b : B$ such that $d \sqsubseteq \beta(b) \ll y$. Moreover, $x \ll d \sqsubseteq \beta(b)$, completing the proof. $\square$

Lemma 4.7.8 (Binary interpolation in the basis for the way-below relation)**.** *If $x \ll z$ and $y \ll z$ in a dcpo D with a small basis $\beta : B \to D$, then there exists an interpolant $b : B$ such that $x, y \ll \beta(b)$ and $\beta(b) \ll z$.*

Proof. The small basis ensures that D is structurally continuous by Lemma 4.7.3. Hence, if $x \ll y$ and $y \ll z$, then there exists an interpolant $d : D$ with $x, y \ll d \ll z$. Now $d \ll z \sqsubseteq \bigsqcup \downarrow_\beta z$, so there exists $b : B$ such that $d \sqsubseteq \beta(b) \ll z$. Moreover, $x, y \ll d \sqsubseteq \beta(b)$, completing the proof. $\square$

Before proving the analogue of Theorem 4.4.12 (closure under Scott continuous retracts) for small bases, we need a type-theoretic analogue of [AJ94, Proposition 2.2.4] and [GHK+03, Proposition III-4.2], which essentially says that it is sufficient for a "subset" of $\downarrow_\beta x$ (given by σ in the lemma) to be directed and have suprema x.

Lemma 4.7.9. *Suppose that we have an element x of a $\mathcal{V}$-dcpo D together with two maps $\beta : B \to D$ and $\sigma : I \to \Sigma_{b:B}(\beta(b) \ll x)$ with $I : \mathcal{V}$. If $\downarrow_\beta x \circ \sigma$ is directed and has supremum x, then $\downarrow_\beta x$ is directed with supremum x too.*

Proof. Suppose that $\downarrow_\beta x \circ \sigma$ is directed and has supremum x. Obviously, x is an upper bound for $\downarrow_\beta x$, so we are to prove that it is the least. If y is an upper bound for $\downarrow_\beta x$, then it is also an upper bound for $\downarrow_\beta x \circ \sigma$ which has supremum x, so that $x \sqsubseteq y$ follows. So the point is directedness of $\downarrow_\beta x$. Its domain is inhabited, because σ is directed. Now suppose that we have $b_1, b_2 : B$ with $\beta(b_1), \beta(b_2) \ll x$. Since $x = \bigsqcup\left(\downarrow_\beta x \circ \sigma\right)$, there exist $i_1, i_2 : I$ such that $\beta(b_1) \sqsubseteq \beta(\mathrm{pr}_1(\sigma(i_1)))$ and $\beta(b_2) \sqsubseteq \beta(\mathrm{pr}_1(\sigma(i_2)))$. Since $\downarrow_\beta x \circ \sigma$ is directed, there exists $i : I$ with $\beta(\mathrm{pr}_1(\sigma(i_1))), \beta(\mathrm{pr}_1(\sigma(i_2))) \sqsubseteq \beta(\mathrm{pr}_1(\sigma(i)))$. Hence, writing $b :\equiv \mathrm{pr}_1(\sigma(i))$, we have $\beta(b) \ll x$ and $\beta(b_1), \beta(b_2) \sqsubseteq \beta(b)$. Thus, $\downarrow_\beta x$ is directed, as desired. □

Theorem 4.7.10. *If we have a Scott continuous retract $D \underset{r}{\overset{s}{\rightleftarrows}} E$ and $\beta : B \to E$ is a small basis for E, then $r \circ \beta$ is a small basis for D.*

Proof. First of all, note that E is locally small by Proposition 4.7.5. But being locally small is closed under Scott continuous retracts by Lemma 3.3.15, so D is locally small too. Moreover, D is structurally continuous by virtue of Theorem 4.4.12 and Lemma 4.7.3. Hence, the way-below relation is small-valued by Proposition 4.4.13. In particular, the type $r(\beta(b)) \ll x$ is small for every $b : B$ and $x : D$.
We are going to use Lemma 4.7.9 to show that $\downarrow_{r\circ\beta} x$ is directed and has supremum x for every $x : D$. By Lemma 4.2.15, the identity on B induces a well-defined map $\sigma : (\Sigma_{b:B}(\beta(b) \ll s(x))) \to (\Sigma_{b:B}(r(\beta(b)) \ll y))$. Now Lemma 4.7.9 tells us that it suffices to prove that $r \circ \downarrow_\beta s(x)$ is directed with supremum x. But $\downarrow_\beta s(x)$ is directed with supremum x, so by Scott continuity of r, the family $r \circ \downarrow_\beta s(x)$ is directed with supremum $r(s(x)) = x$, as desired. □

Finally, a nice use of dcpos with small bases is that they yield locally small exponentials, as we can restrict the quantification in the pointwise order to elements of the small basis.

Proposition 4.7.11. *If D is a dcpo with an unspecified small basis and E is a locally small dcpo, then the exponential E^D is locally small too.*

Proof. Being locally small is a proposition, so in proving the result we may assume that D comes equipped with a small basis $\beta : B \to D$. For arbitrary Scott continuous functions $f, g : D \to E$, we claim that $f \sqsubseteq g$ precisely when $\forall_{b:B}(f(\beta(b)) \sqsubseteq g(\beta(b)))$, which is a small type using that E is locally small. The left-to-right implication is obvious, so suppose that $f(\beta(b)) \sqsubseteq g(\beta(b))$ for every $b : B$ and let $x : D$ be arbitrary. We are to show that $f(x) \sqsubseteq g(x)$. Since $x = \bigsqcup \mathord{\downarrow\!\!\downarrow}_\beta x$, it suffices to prove $f\left(\bigsqcup \mathord{\downarrow\!\!\downarrow}_\beta x\right) \sqsubseteq g\left(\bigsqcup \mathord{\downarrow\!\!\downarrow}_\beta x\right)$ and in turn, that $f(\beta(b)) \sqsubseteq g\left(\bigsqcup \mathord{\downarrow\!\!\downarrow}_\beta x\right)$ for every $b : B$. But is easily seen to hold, because $f(\beta(b)) \sqsubseteq g(\beta(b))$ for every $b : B$ by assumption. ☐

4.8 Small compact bases

Similarly to the progression from continuous dcpos (Section 4.4) to algebraic ones (Section 4.6), we now turn to small *compact* bases.

Definition 4.8.1 (Small compact basis)**.** For a $\mathcal{V}$-dcpo D, a map $\beta : B \to D$ with $B : \mathcal{V}$ is a *small compact basis* for D if the following conditions hold:

(i) for every $b : B$, the element $\beta(b)$ is compact in D;

(ii) for every $x : D$, the family $(\Sigma_{b:B}(\beta(b) \sqsubseteq x)) \xrightarrow{\beta \circ \mathrm{pr}_1} D$ is directed and has supremum x;

(iii) for every $x : D$ and $b : B$, the proposition $\beta(b) \sqsubseteq x$ is $\mathcal{V}$-small.

We will write $\downarrow_\beta x$ for the type $\Sigma_{b:B}(\beta(b) \sqsubseteq x)$ and conflate this type with the canonical map $\downarrow_\beta x \xrightarrow{\beta \circ \mathrm{pr}_1} D$.

Remark 4.8.2. If $\beta : B \to D$ is a small compact basis for a $\mathcal{V}$-dcpo D, then the type $\downarrow_\beta x$ is small. Similarly to Remark 4.7.2, we will use this tacitly and write as if the type $\downarrow_\beta x$ is actually a type in $\mathcal{V}$.

Lemma 4.8.3. *If a dcpo comes equipped with a small compact basis, then it is structurally algebraic. Hence, if a dpco has an unspecified small compact basis, then it is algebraic.*

Proof. For every element x of a dcpo D, the family $\downarrow_\beta x \to D$ consists of compact elements and approximates x, so the assignment $x \mapsto \downarrow_\beta x$ makes D structurally continuous. ☐

Lemma 4.8.4. *A map $\beta : B \to D$ is a small compact basis for a dcpo D if and only if β is a small basis for D and $\beta(b)$ is compact for every $b : B$.*

Proof. If $\beta(b)$ is compact for every $b : B$, then $\beta(b) \sqsubseteq x$ if and only if $\beta(b) \ll x$ for every $b : B$ and $x : D$, so that $\mathord{\downarrow\!\!\downarrow}_\beta x \simeq \downarrow_\beta x$ for every $x : D$. In particular, $\mathord{\downarrow\!\!\downarrow}_\beta x$ approximates x if and only if $\downarrow_\beta x$ does, which completes the proof. ☐

Proposition 4.8.5. *A small compact basis contains every compact element. That is, if $\beta : B \to D$ is a small compact basis for a dcpo D and $x : D$ is compact, then there exists $b : B$ such that $\beta(b) = x$.*

Proof. Suppose we have a compact element $x : D$. By compactness of x and the fact that $x = \downarrow_\beta x$, there exists $b : B$ with $\beta(b) \ll x$ such that $x \sqsubseteq \beta(b)$. But then $\beta(b) = x$ by antisymmetry. □

4.9 Examples of dcpos with small compact bases

Armed with the theory of small bases we turn to examples illustrating small bases in practice. Our examples will involve small *compact* bases and an example of a dcpo with a small basis that is not compact will have to wait until Section 4.10.2 when we have developed the ideal completion.

Example 4.9.1. The map $\beta : \mathbf{2} \to \Omega_{\mathcal{U}}$ defined by $0 \mapsto \mathbf{0}_{\mathcal{U}}$ and $1 \mapsto \mathbf{1}_{\mathcal{U}}$ is a small compact basis for $\Omega_{\mathcal{U}}$. In particular, $\Omega_{\mathcal{U}}$ is (structurally) algebraic.

The basis $\beta : \mathbf{2} \to \Omega_{\mathcal{U}}$ defined above has the special property that it is *dense* in the sense of [Esc+, TypeTopology.Density]: its image has empty complement, i.e. the type $\Sigma_{P:\Omega_{\mathcal{U}}} \neg(\Sigma_{b:\mathbf{2}}\, \beta(b) = P)$ is empty.

Proof of Example 4.9.1. By Example 4.2.5, every element in the image of β is compact. Moreover, $\Omega_{\mathcal{U}}$ is locally small, so we only need to prove that for every $P : \Omega_{\mathcal{U}}$ the family $\downarrow_\beta P$ is directed with supremum P. The domain of the family is inhabited, because $\beta(0)$ is the least element. Semidirectedness also follows easily, since $\mathbf{2}$ has only two elements for which we have $\beta(0) \sqsubseteq \beta(1)$. Finally, the supremum of $\downarrow_\beta P$ is obviously below P. Conversely, if P holds, then $\bigsqcup \downarrow_\beta P = \mathbf{1} = P$. The final claim follows from Lemma 4.8.3. □

Example 4.9.2. For a set $X : \mathcal{U}$, the map $\beta : (\mathbf{1} + X) \to \mathcal{L}_{\mathcal{U}}(X)$ given by $\mathrm{inl}(\star) \mapsto \bot$ and $\mathrm{inr}(x) \mapsto \eta(x)$ is a small compact basis for $\mathcal{L}_{\mathcal{U}}(X)$. In particular, $\mathcal{L}_{\mathcal{U}}(X)$ is (structurally) algebraic.

Similar to Example 4.9.1, the basis $\beta : (\mathbf{1} + X) \to \mathcal{L}_{\mathcal{U}}(X)$ defined above is also dense.

Proof of Example 4.9.2. By Example 4.2.6, every element in the image of β is compact. Moreover, the lifting is locally small by Proposition 3.4.14, so we only need to prove that for every partial element l, the family $\downarrow_\beta l$ is directed with supremum l. The domain of the family is inhabited, because $\beta(\mathrm{inl}(\star))$ is the least element. Semidirectedness also follows easily: First of all, $\beta(\mathrm{inl}(\star))$ is the least element. Secondly, if we have $x, x' : X$ such that $\beta(\mathrm{inr}(x)), \beta(\mathrm{inr}(x')) \sqsubseteq l$, then because $\beta(\mathrm{inr}(x)) \equiv \eta(x)$ is defined, we must have $\beta(\mathrm{inr}(x)) = l = \beta(\mathrm{inr}(x'))$ by definition of the order. Finally, the supremum of $\downarrow_\beta l$ is obviously a partial element below l. Conversely, if l is defined, then $l = \eta(x)$ for some $x : X$, and hence, $l = \eta(x) \sqsubseteq \bigsqcup \downarrow_\beta l$. The final claim follows from Lemma 4.8.3. □

Example 4.9.3. For a set $X : \mathcal{U}$, the map $\beta : \mathrm{List}(X) \to \mathcal{P}_{\mathcal{U}}(X)$ from Definition 4.2.12 (whose image is the type of Kuratowski finite subsets of X) is a small compact basis for $\mathcal{P}_{\mathcal{U}}(X)$. In particular, $\mathcal{P}_{\mathcal{U}}(X)$ is (structurally) algebraic.

Notice that the map $\beta : \mathrm{List}(X) \to \mathcal{P}(X)$ is not an embedding, as two lists can represent the same Kuratowski finite subset. Of course, an embedding is given by the inclusion of the Kuratowski finite subsets into the powerset, and its codomain is small if we assume set replacement, because it is the image of β.

Proof of Example 4.9.3. By Lemma 4.2.13 and Example 4.2.14, all elements in the image of β are compact. Moreover, $\mathcal{P}_{\mathcal{U}}(X)$ is locally small, so we only need to prove that for every $A : \mathcal{P}(X)$ the family $\downarrow_\beta A$ is directed with supremum A, but this was also proven in Example 4.2.14. The final claim follows from Lemma 4.8.3. □

At this point the reader may ask whether we have any examples of dcpos that are structurally algebraic but that do not have a small compact basis. The following example shows that this can happen in our predicative setting.

Example 4.9.4. The lifting $\mathcal{L}_{\mathcal{V}}(P)$ of a proposition $P : \mathcal{U}$ is structurally algebraic, but has a small compact basis if and only if P is $\mathcal{V}$-small.

Thus, requiring that $\mathcal{L}_{\mathcal{V}}(P)$ has a small basis for every proposition $P : \mathcal{U}$ is equivalent to Propositional-Resizing$_{\mathcal{U},\mathcal{V}}$.

Proof of Example 4.9.4. Note that $\mathcal{L}_{\mathcal{V}}(P)$ is simply the type of propositions in $\mathcal{V}$ that imply P. It is structurally algebraic, because given such a proposition Q, the family

$$\begin{aligned} Q + \mathbf{1}_{\mathcal{V}} &\to \mathcal{L}_{\mathcal{V}}(P) \\ \mathrm{inl}(q) &\mapsto \mathbf{1}_{\mathcal{V}} \\ \mathrm{inr}(\star) &\mapsto \mathbf{0}_{\mathcal{V}} \end{aligned}$$

is directed, has supremum Q and consists of compact elements. But if $\mathcal{L}_{\mathcal{V}}(P)$ had a small compact basis $\beta : B \to D$, then we would have $P \simeq \exists_{b:B}(\beta(b) \simeq \mathbf{1}_{\mathcal{V}})$ and the latter is $\mathcal{V}$-small. □

4.10 The rounded ideal completion

We have seen that in continuous dcpos, the basis essentially "generates" the whole dcpo, because the basis suffices to approximate any of its elements. It is natural to ask whether one can start from a more abstract notion of basis and "complete" it to a continuous dcpo. This is exactly what we do here using the notion of an *abstract basis* and the *rounded ideal completion*.

Definition 4.10.1 (Abstract basis)**.** An *abstract $\mathcal{V}$-basis* is a type $B : \mathcal{V}$ with a binary relation $\prec : B \to B \to \mathcal{V}$ that is proposition-valued, transitive and satisfies
nullary interpolation: for every $a : B$, there exists $b : B$ with $b \prec a$, and
binary interpolation: for every $a_1, a_2 \prec b$, there exists $a : B$ with $a_1, a_2 \prec a \prec b$.

Definition 4.10.2 (Ideal, (rounded) ideal completion, $\mathcal{V}$-Idl$(B, \prec)$)**.**
(i) A subset $I : B \to \Omega_{\mathcal{V}}$ of an abstract $\mathcal{V}$-basis $(B, \prec)$ is a *$\mathcal{V}$-ideal* if it is a directed

lower set with respect to $\prec$. That it is a lower set means: if $b \in I$ and $a \prec b$, then $a \in I$ too.

(ii) We write $\mathcal{V}$-Idl$(B, \prec)$ for the type of $\mathcal{V}$-ideals of an abstract $\mathcal{V}$-basis $(B, \prec)$ and call $\mathcal{V}$-Idl$(B, \prec)$ the *(rounded) ideal completion* of $(B, \prec)$.

The name rounded ideal completion is justified by Lemmas 4.10.5 and 4.10.6 below.

Definition 4.10.3 (Union of ideals, $\bigcup \mathcal{I}$)**.** Given a family $\mathcal{I} : S \to \mathcal{V}\text{-Idl}(B, \prec)$ of ideals, indexed by $S : \mathcal{V}$, we write

$$\bigcup \mathcal{I} :\equiv \{b \in B \mid \exists_{s:S}(b \in \mathcal{I}_s)\}$$

for the set-theoretic union of the ideals indexed by $\mathcal{I}$.

Lemma 4.10.4. *If $\mathcal{I} : S \to \mathcal{V}\text{-Idl}(B, \prec)$ is directed, then $\bigcup \mathcal{I}$ is an ideal.*

Proof. The subset $\bigcup \mathcal{I}$ is easily seen to be a lower set, for if $a \prec b \in \bigcup \mathcal{I}$, then there exists $s : S$ such that $a \prec b \in \mathcal{I}_s$, so $a \in \mathcal{I}_s$ as $\mathcal{I}_s$ is a lower set, but then $a \in \bigcup \mathcal{I}$. Moreover, $\bigcup \mathcal{I}$ is inhabited: Since $\mathcal{I}$ is directed, there exists $s : S$, but $\mathcal{I}_s$ is an ideal and therefore inhabited, so there exists $b \in \mathcal{I}_s$ which implies $b \in \bigcup \mathcal{I}$. Finally, suppose we have $b_1, b_2 \in \bigcup \mathcal{I}$. By definition, there exist $s_1, s_2 : S$ such that $b_1 \in \mathcal{I}_{s_1}$ and $b_2 \in \mathcal{I}_{s_2}$. By directedness of S, there exists $s : S$ such that $\mathcal{I}_{s_1}, \mathcal{I}_{s_2} \subseteq \mathcal{I}_s$. Hence, $b_1, b_2 \in \mathcal{I}_s$, which is an ideal, so there exists $b \in \mathcal{I}_s$ with $b_1, b_2 \prec b$. But then also $b \in \bigcup \mathcal{I}$, which proves that $\bigcup \mathcal{I}$ is directed and hence an ideal, completing the proof. □

Lemma 4.10.5. *The rounded ideal completion of an abstract $\mathcal{V}$-basis $(B, \prec)$ is a $\mathcal{V}$-dcpo when ordered by subset inclusion.*

Proof. Since taking unions yields the least upper bound in the powerset, we only have to prove that the union of ideals is again an ideal, but this is Lemma 4.10.4. □

Paying attention to the universe levels, the ideals form a large but locally small $\mathcal{V}$-dcpo because $\mathcal{V}\text{-Idl}(B, \prec) : \mathcal{V}\text{-DCPO}_{\mathcal{V}^+,\mathcal{V}}$. For the remainder of this section, we will fix an abstract $\mathcal{V}$-basis $(B, \prec)$ and consider its $\mathcal{V}$-ideals.

Lemma 4.10.6 (Roundedness)**.** *The ideals of an abstract basis are* rounded*: for every element a of an ideal I, there exists $b \in I$ such that $a \prec b$.*

Proof. Because ideals are semidirected. □

Roundedness makes up for the fact that we have not required an abstract basis to be reflexive. If it is, then (Section 4.10.1) the ideal completion is structurally algebraic.

Definition 4.10.7 (Principal ideal, $\downarrow b$)**.** The *principal ideal* of an element $b : B$ is defined as the subset $\downarrow b :\equiv \{a \in B \mid a \prec b\}$. Observe that the principal ideal is indeed an ideal: it is a lower set by transitivity of $\prec$, and inhabited and semidirected precisely by nullary and binary interpolation, respectively.

Lemma 4.10.8. *The assignment $b \mapsto \downarrow b$ is monotone, i.e. if $a \prec b$, then $\downarrow a \subseteq \downarrow b$.*

Proof. By transitivity of $\prec$. □

Lemma 4.10.9. *Every ideal is the directed supremum of its principal ideals. That is, for an ideal I, the family $(\Sigma_{b:B}(b \in I)) \xrightarrow{b \mapsto \downarrow b} \mathcal{V}\text{-Idl}(B, \prec)$ is directed and has supremum I.*

Proof. Since ideals are lower sets, we have $\downarrow b \subseteq I$ for every $b \in I$. Hence, the union $\bigcup_{b \in I} \downarrow b$ is a subset of I. Conversely, if $a \in I$, then by roundedness of I there exists $a' \in I$ with $a \prec a'$, so that $a \in \bigcup_{b \in I} \downarrow b$. So it remains to show that the family is directed. Notice that it is inhabited, because I is an ideal. Now suppose that $b_1, b_2 \in I$. Since I is directed, there exists $b \in I$ such that $b_1, b_2 \prec b$. But this implies $\downarrow b_1, \downarrow b_2 \subseteq \downarrow b$ by Lemma 4.10.8, so the family is semidirected, as desired. □

Lemma 4.10.10. *The following are equivalent for every two ideals I and J:*

(i) $I \ll J$;
(ii) there exists $b \in J$ such that $I \subseteq \downarrow b$;
(iii) there exist $a \prec b$ such that $I \subseteq \downarrow a \subseteq \downarrow b \subseteq J$.

In particular, if b is an element of an ideal I, then $\downarrow b \ll I$.

Proof. We show that (i) ⇒ (ii) ⇒ (iii) ⇒ (i). So suppose that $I \ll J$. Then J is the directed supremum of its principal ideals by Lemma 4.10.9. Hence, there exists $b \in J$ such that $I \subseteq \downarrow b$ already, which is exactly (ii). Now suppose that we have $a \in J$ with $I \subseteq \downarrow a$. By roundedness of J, there exists $b \in J$ with $a \prec b$. But then $I \subseteq \downarrow a \subseteq \downarrow b \subseteq J$ by Lemma 4.10.8 and the fact that J is a lower set, establishing (iii). Now suppose that condition (iii) holds and that J is a subset of some directed join of ideals $\mathcal{J}$ indexed by a type $S : \mathcal{V}$. Since $a \in \downarrow b \subseteq J$, there exists $s : S$ such that $a \in \mathcal{J}_s$. In particular, $\downarrow a \subseteq \mathcal{J}_s$ because ideals are lower sets. Hence, if $a' \in I \subseteq \downarrow a$, then $a' \in \mathcal{J}_s$, so $I \subseteq \mathcal{J}_s$, which proves that $I \ll J$.
Finally, if b is an element of an ideal I, then $\downarrow b \ll I$, because (ii) implies (i) and $\downarrow b \subseteq \downarrow b$ obviously holds. □

Theorem 4.10.11. *The principal ideals $\downarrow(-) : B \to \mathcal{V}\text{-Idl}(B, \prec)$ yield a small basis for $\mathcal{V}\text{-Idl}(B, \prec)$. In particular, $\mathcal{V}\text{-Idl}(B, \prec)$ is (structurally) continuous.*

Proof. First of all, note that the way-below relation on $\mathcal{V}\text{-Idl}(B, \prec)$ is small-valued because of Lemma 4.10.10. So it remains to show that for every ideal I, the family $(\Sigma_{b:B}(\downarrow b \ll I)) \xrightarrow{b \mapsto \downarrow b} \mathcal{V}\text{-Idl}(B, \prec)$ is directed with supremum I. That the domain of this family is inhabited follows from Lemma 4.10.10 and the fact that I is inhabited. For semidirectedness, suppose we have $b_1, b_2 : B$ with $\downarrow b_1, \downarrow b_2 \ll I$. By Lemma 4.10.10 there exist $c_1, c_2 \in I$ such that $\downarrow b_1 \subseteq \downarrow c_1$ and $\downarrow b_2 \subseteq \downarrow c_2$. Since I is directed, there exists $b \in I$ with $c_1, c_2 \prec b$. But now $\downarrow b_1 \subseteq \downarrow c_1 \subseteq \downarrow b \ll I$ by Lemmas 4.10.8 and 4.10.10 and similarly, $\downarrow b_2 \subseteq \downarrow b \ll I$. Hence, the family is semidirected, as we wished to show. Finally, we show that I is the supremum of the family. If $b \in I$, then, since I is rounded, there exists $c \in I$ with $b \prec c$. Moreover, ${\downarrow} c \ll I$ by Lemma 4.10.10. Hence, b is included in the join of the family. Conversely, if we have $b : B$ with ${\downarrow} b \ll I$, then ${\downarrow} b \subseteq I$, so I is also an upper bound for the family. ☐

4.10.1 The rounded ideal completion of a reflexive abstract basis

Lemma 4.10.12. *If $\prec : B \to B \to \mathcal{V}$ is proposition-valued, transitive and reflexive, then $(B, \prec)$ is an abstract basis.*

Proof. The interpolation properties for $\prec$ are easily proved when it is reflexive. ☐

Lemma 4.10.13. *If an element $b : B$ is reflexive, i.e. $b \prec b$ holds, then $b \in I$ if and only if ${\downarrow} b \subseteq I$ for every ideal I.*

Proof. The left-to-right implication holds because I is a lower set and the converse holds because $b \in {\downarrow} b$ as b is assumed to be reflexive. ☐

Lemma 4.10.14. *If $b : B$ is reflexive, then its principal ideal ${\downarrow} b$ is compact.*

Proof. Suppose that we have $b : B$ such that $b \prec b$ holds and that ${\downarrow} b \subseteq \bigcup \mathcal{I}$ for some directed family $\mathcal{I}$ of ideals. By Lemma 4.10.13, we have $b \in \bigcup \mathcal{I}$, which means that there exists s in the domain of $\mathcal{I}$ such that $b \in \mathcal{I}_s$. Using Lemma 4.10.13 once more, we see that ${\downarrow} b \subseteq \mathcal{I}_s$, proving that ${\downarrow} b$ is compact. ☐

Theorem 4.10.15. *If $\prec$ is reflexive, then the principal ideals ${\downarrow}(-) : B \to \mathcal{V}\text{-Idl}(B, \prec)$ yield a small* compact *basis for $\mathcal{V}\text{-Idl}(B, \prec)$. In particular, $\mathcal{V}\text{-Idl}(B, \prec)$ is (structurally) algebraic.*

Proof. This follows from Theorem 4.10.11 and Lemmas 4.8.4 and 4.10.14. ☐

Theorem 4.10.16. *If $f : B \to D$ is a monotone map to a $\mathcal{V}$-dcpo D, then the map $\bar{f} : \mathcal{V}\text{-Idl}(B, \prec) \to D$ defined by taking an ideal I to the supremum of the directed family $f \circ \mathrm{pr}_1 : (\Sigma_{b:B}(b \in I)) \to D$ is Scott continuous. Moreover, if $\prec$ is reflexive, then the diagram*

$$\begin{tikzcd} B \arrow[rr, "f"] \arrow[dr, "{\downarrow}(-)"'] & & D \\ & \mathcal{V}\text{-Idl}(B, \prec) \arrow[ur, "\bar{f}"'] & \end{tikzcd}$$

commutes.

Proof. Note that $f \circ \mathrm{pr}_1 : (\Sigma_{b:B}(b \in I)) \to D$ is indeed a directed family, because I is a directed subset of B and f is monotone. For Scott continuity of $\bar{f}$, assume that

we have a directed family $\mathcal{I}$ of ideals indexed by $S : \mathcal{V}$. We first show that $\bar{f}(\bigcup \mathcal{I})$ is an upper bound of $\bar{f} \circ \mathcal{I}$. So let $s : S$ be arbitrary and note that $\bar{f}(\bigcup \mathcal{I}) \sqsupseteq \bar{f}(\mathcal{I}_s)$ as soon as $\bar{f}(\bigcup \mathcal{I}) \sqsupseteq f(b)$ for every $b \in \mathcal{I}_s$. But for such b we have $b \in \bigcup \mathcal{I}$, so this holds. Now suppose that y is an upper bound of $\bar{f} \circ \mathcal{I}$. To show that $\bar{f}(\bigcup \mathcal{I}) \sqsubseteq y$, it is enough to prove that $f(b) \sqsubseteq y$ for every $b \in \mathcal{I}$. But for such b, there exists $s : S$ such that $b \in \mathcal{I}_s$ and hence, $f(b) \sqsubseteq \bar{f}(\mathcal{I}_s) \sqsubseteq y$.
Finally, if $\prec$ is reflexive, then we prove that $\bar{f}(\downarrow b) = f(b)$ for every $b : B$ by antisymmetry. Since $\prec$ is assumed to be reflexive, we have $b \in \downarrow b$ and therefore, $f(b) \sqsubseteq \bar{f}(\downarrow b)$. Conversely, for every $c \prec b$ we have $f(c) \sqsubseteq f(b)$ by monotonicity of f and hence, $\bar{f}(\downarrow b) \sqsubseteq f(b)$, as desired. □

4.10.2 Example: the ideal completion of dyadics

We end this section by describing an example of a continuous dcpo, built using the ideal completion, that is not algebraic. In fact, this dcpo has no compact elements at all.

We inductively define a type and an order representing dyadic rationals $m/2^n$ in the interval $(-1, 1)$ for integers m, n. This type is similar to the lower Dedekind reals but with dyadics instead of rationals and is extended with a point at $+\infty$. We prefer to work with this type, because working with lower Dedekind reals would require us to develop and formalise the theory of integers, rational numbers, etc.

The intuition for the upcoming definitions is the following. Start with the point 0 in the middle of the interval (represented by *middle* below). Then consider the two functions (respectively represented by *left* and *right* below)

$$\begin{aligned} l, r &: (-1, 1) \to (-1, 1) \\ l(x) &:\equiv (x-1)/2 \\ r(x) &:\equiv (x+1)/2 \end{aligned}$$

that generate the dyadic rationals. Observe that $l(x) < 0 < r(x)$ for every $x : (-1, 1)$. Accordingly, we inductively define the following types.

Definition 4.10.17 (Dyadics, $\mathbb{D}$, $\prec$)**.** The type of *dyadics* $\mathbb{D} : \mathcal{U}_0$ is the inductive type with these three constructors

$$\text{middle} : \mathbb{D} \qquad \text{left} : \mathbb{D} \to \mathbb{D} \qquad \text{right} : \mathbb{D} \to \mathbb{D}.$$

We also inductively define $\prec : \mathbb{D} \to \mathbb{D} \to \mathcal{U}_0$ as

$$\begin{array}{llllll} \text{middle} \prec \text{middle} & :\equiv \mathbf{0} & \text{left}(x) \prec \text{middle} & :\equiv \mathbf{1} & \text{right}(x) \prec \text{middle} & :\equiv \mathbf{0} \\ \text{middle} \prec \text{left}(y) & :\equiv \mathbf{0} & \text{left}(x) \prec \text{left}(y) & :\equiv x \prec y & \text{right}(x) \prec \text{left}(y) & :\equiv \mathbf{0} \\ \text{middle} \prec \text{right}(y) & :\equiv \mathbf{1} & \text{left}(x) \prec \text{right}(y) & :\equiv \mathbf{1} & \text{right}(x) \prec \text{right}(y) & :\equiv x \prec y. \end{array}$$

Lemma 4.10.18. *The type of dyadics is a set with decidable equality.*

Proof. Sethood follows from having decidable equality by Hedberg's Theorem. To see that $\mathbb{D}$ has decidable equality, one can use a standard inductive proof. $\square$

Definition 4.10.19 (Trichotomy, density, having no endpoints)**.** We say that a binary relation $<$ on a type X is

- *trichotomous* if exactly one of $x \prec y$, $x = y$ or $y \prec x$ holds.
- *dense* if for every $x, y : X$, there exists some $z : X$ such that $x \prec z \prec y$.
- *without endpoints* if for every $x : X$, there exist some $y, z : X$ with $y \prec x \prec z$.

Lemma 4.10.20. *The relation $\prec$ on the dyadics is proposition-valued, transitive, irreflexive, trichotomous, dense and without endpoints.*

Proof. That $\prec$ is proposition-valued, transitive, irreflexive and trichotomous is all proven by a straightforward induction on the definition on $\mathbb{D}$. That it has no endpoints is witnessed by the fact that for every $x : \mathbb{D}$, we have

$$\operatorname{left} x \prec x \prec \operatorname{right} x \tag{†}$$

which is proven by induction on $\mathbb{D}$ as well. We spell out the inductive proof that it is dense, making use of (†). Suppose that $x \prec y$. Looking at the definition of the order, we see that we need to consider five cases.

- If $x = \text{middle}$ and $y = \operatorname{right} y'$, then we have $x \prec \operatorname{right}(\operatorname{left}(y')) \prec y$.
- If $x = \operatorname{left}(x')$ and $y = \text{middle}$, then we have $x \prec \operatorname{left}(\operatorname{right}(x')) \prec y$.
- If $x = \operatorname{left}(x')$ and $y = \operatorname{right} y'$, then we have $x \prec \text{middle} \prec y$.
- If $x = \operatorname{right}(x')$ and $y = \operatorname{right} y'$, then we have $x' \prec y'$ and therefore, by induction hypothesis, there exists $z' : \mathbb{D}$ such that $x' \prec z' \prec y'$. Hence, $x \prec \operatorname{right}(z') \prec y$.
- If $x = \operatorname{left}(x')$ and $y = \operatorname{left}(y')$, then $x' \prec y'$ and hence, by induction hypothesis, there exists $z' : \mathbb{D}$ such that $x' \prec z' \prec y'$. Thus, $x \prec \operatorname{left}(z') \prec y$. $\square$

Proposition 4.10.21. *The pair $(\mathbb{D}, \prec)$ is an abstract $\mathcal{U}_0$-basis.*

Proof. By Lemma 4.10.20 the relation $\prec$ is proposition-valued and transitive. Moreover, that it has no endpoints implies unary interpolation. For binary interpolation, suppose that we have $x \prec z$ and $y \prec z$. Then by trichotomy there are three cases.

- If $x = y$, then using density and our assumption that $x \prec z$, there exists $d : \mathbb{D}$ with $y = x \prec d \prec z$, as desired.
- If $x \prec y$, then using density and our assumption that $y \prec z$, there exists $d : \mathbb{D}$ with $y \prec d \prec z$, but then also $x \prec d$ since $x \prec y$, so we are done.
- If $x \prec y$, then the proof is similar to that of the second case. $\square$

Proposition 4.10.22. *The ideal completion $\mathcal{U}_0$-$\mathrm{Idl}(\mathbb{D}, \prec) : \mathcal{U}_0$-$\mathrm{DCPO}_{\mathcal{U}_1,\mathcal{U}_0}$ is structurally continuous with small basis $\downarrow(-) : \mathbb{D} \to \mathcal{U}_0$-$\mathrm{Idl}(\mathbb{D}, \prec)$. Moreover, it cannot be algebraic, because none of its elements are compact.*

Proof. The first claim follows from Theorem 4.10.11. Now suppose for a contradiction that we have a compact ideal I. By Lemma 4.10.10, there exists $x \in I$ with $I \subseteq \downarrow x$. But this implies $x \prec x$, which is impossible as $\prec$ is irreflexive. $\square$

4.11 Ideal completions of small bases

Given a $\mathcal{V}$-dcpo D with a small basis $\beta : B \to D$, we show that there are two natural ways of turning B into an abstract basis. Either define $b \prec c$ by $\beta(b) \ll \beta(c)$, or by $\beta(b) \sqsubseteq \beta(c)$. Taking their $\mathcal{V}$-ideal completions we show that the former yields a continuous dcpo isomorphic to D, while the latter yields an algebraic dcpo (with a small compact basis) in which D can be embedded. The latter fact will find application in Section 4.13, while the former gives us a presentation theorem: every dcpo with a small basis is isomorphic to a dcpo of ideals. In particular, if $D : \mathcal{V}\text{-DCPO}_{\mathcal{U},\mathcal{T}}$ has a small basis, then it is isomorphic to a dcpo with simpler universe parameters, namely $\mathcal{V}\text{-Idl}(B, \ll_\beta) : \mathcal{V}\text{-DCPO}_{\mathcal{V}^+,\mathcal{V}}$. Of course a similar result holds for dcpos with a small compact basis. In studying these variations, it is helpful to first develop some machinery that all of them have in common.

Fix a $\mathcal{V}$-dcpo D with a small basis $\beta : B \to D$. In what follows we conflate the family $\downarrow_\beta x : (\Sigma_{b:B}(\beta(b) \ll x)) \xrightarrow{\beta \circ \mathrm{pr}_1} D$ with its associated subset $\{b \in B \mid \beta(b) \ll x\}$, formally given by the map $B \to \Omega_{\mathcal{V}}$ defined as $b \mapsto \exists_{b:B}(\beta(b) \ll x)$.

Lemma 4.11.1. *The assignment $x : D \mapsto \downarrow_\beta x : \mathcal{P}(B)$ is Scott continuous.*

Proof. Note that $\downarrow_\beta(-)$ is monotone: if $x \sqsubseteq y$ and $b : B$ is such that $\beta(b) \ll x$, then also $\beta(b) \ll y$. So it suffices to prove that $\downarrow_\beta(\bigsqcup \alpha) \subseteq \bigcup_{i:I} \downarrow_\beta \alpha_i$. So suppose that $b : B$ is such that $\beta(b) \ll \bigsqcup \alpha$. By Lemma 4.7.7, there exists $c : B$ with $\beta(b) \ll \beta(c) \ll \bigsqcup \alpha$. Hence, there exists $i : I$ such that $\beta(b) \ll \beta(c) \sqsubseteq \alpha_i$ already, and therefore, $b \in \bigcup_{j:J} \downarrow_\beta \alpha_j$, as desired. $\square$

By virtue of the fact that β is a small basis for D, we know that taking the directed supremum of $\downarrow_\beta x$ equals x for every $x : D$. In other words, $\downarrow_\beta(-)$ is a section of $\bigsqcup(-)$. The following lemma gives conditions for the other composite to be an inflation or a deflation.

Lemma 4.11.2. *Let $I : B \to \Omega_{\mathcal{V}}$ be a subset of B such that its associated family $\bar{I} : (\Sigma_{b:B}(b \in I)) \xrightarrow{\beta \circ \mathrm{pr}_1} D$ is directed.*

(i) If the conjunction of $\beta(b) \sqsubseteq \beta(c)$ and $c \in I$ implies $b \in I$, then $\downarrow_\beta \bigsqcup \bar{I} \subseteq I$.

(ii) If for every $b \in I$ there exists $c \in I$ such that $\beta(b) \ll \beta(c)$, then $I \subseteq \downarrow_\beta \bigsqcup \bar{I}$.

In particular, if both conditions hold, then $I = \downarrow_\beta \bigsqcup \bar{I}$.

Note that the first condition says that I is a lower set with respect to the order of D, while the second says that I is rounded with respect to the way-below relation.

Proof. (i) Suppose that I is a lower set and let $b : B$ be such that $\beta(b) \ll \bigsqcup \bar{I}$. Then there exists $c \in I$ with $\beta(b) \sqsubseteq \beta(c)$, which implies $b \in I$ as desired, because I is assumed to be a lower set. (ii) Assume that I is rounded and let $b \in I$ be arbitrary. By roundedness of I, there exists $c \in I$ such that $\beta(b) \ll \beta(c)$. But then $\beta(b) \ll \beta(c) \sqsubseteq \bigsqcup \bar{I}$, so that $b \in \downarrow_\beta \bigsqcup \bar{I}$, as we wished to show. □

Lemma 4.11.3. *Suppose that we have $\prec : B \to B \to \mathcal{V}$ and let $x : D$ be arbitrary.*

(i) *If $b \prec c$ implies $\beta(b) \sqsubseteq \beta(c)$ for every $b, c : B$, then $\downarrow_\beta x$ is a lower set w.r.t. $\prec$.*

(ii) *If $\beta(b) \ll \beta(c)$ implies $b \prec c$ for every $b, c : B$, then $\downarrow_\beta x$ is semidirected w.r.t. $\prec$.*

Proof. (i) This is immediate, because $\downarrow \beta x$ is a lower set with respect to the order relation on D. (ii) Suppose that the condition holds and that we have $b_1, b_2 : B$ such that $\beta(b_1), \beta(b_2) \ll x$. Using binary interpolation in the basis, there exist $c_1, c_2 : B$ with $\beta(b_1) \ll \beta(c_1) \ll x$ and $\beta(b_2) \ll \beta(c_2) \ll x$. Hence, $c_1, c_2 \in \downarrow_{\beta x}$ and moreover, by assumption we have $b_1 \prec c_1$ and $b_2 \prec c_2$, as desired. □

4.11.1 Ideal completion with respect to the way-below relation

Lemma 4.11.4. *If $\beta : B \to D$ is a small basis for a $\mathcal{V}$-dcpo D, then $(B, \ll_\beta)$ is an abstract $\mathcal{V}$-basis where $b \ll_\beta c$ is defined as $\beta(b) \ll \beta(c)$.*

Remark 4.11.5. The definition of an abstract $\mathcal{V}$-basis requires the relation on it to be $\mathcal{V}$-valued. Hence, for the lemma to make sense we appeal to the fact that β is a *small* basis which tells us that we can substitute $\beta(b) \ll \beta(c)$ by an equivalent type in $\mathcal{V}$.

Proof of Lemma 4.11.4. The way-below relation is proposition-valued and transitive. Moreover, $\ll_\beta$ satisfies nullary and binary interpolation precisely because we have nullary and binary interpolation in the basis for the way-below relation by Lemmas 4.7.6 and 4.7.8. □

The following theorem is a presentation result for dcpos with a small basis: every such dcpo can be presented as the rounded ideal completion of its small basis.

Theorem 4.11.6. *The map $\downarrow_\beta (-) : D \to \mathcal{V}\text{-Idl}(B, \ll_\beta)$ is an isomorphism of $\mathcal{V}$-dcpos.*

Proof. First of all, we should check that the map is well-defined, i.e. that $\downarrow_\beta x$ is an $(B, \ll_\beta)$-ideal. It is an inhabited subset by nullary interpolation in the basis and a semidirected lower set because the criteria of Lemma 4.11.3 are satisfied when taking $\prec$ to be $\ll_\beta$. Secondly, the map $\downarrow_\beta (-)$ is Scott continuous by Lemma 4.11.1.

Now notice that the map $\beta : (B, \ll_\beta) \to D$ is monotone and that the Scott continuous map it induces by Theorem 4.10.16 is exactly the map $\bigsqcup : \mathcal{V}\text{-Idl}(B, \ll_\beta) \to D$ that takes an ideal I to the supremum of its associated directed family $\beta \circ \mathrm{pr}_1 : (\Sigma_{b:B}(b \in I)) \to D$.

Since β is a basis for D, we know that $\bigsqcup \downarrow_\beta x = x$ for every $x : D$. So it only remains to show that $\downarrow_\beta \circ \bigsqcup$ is the identity on $\mathcal{V}\text{-Idl}(B, \ll_\beta)$, for which we will use Lemma 4.11.2. So suppose that $I : \mathcal{V}\text{-Idl}(B, \ll_\beta)$ is arbitrary. Then we only need to prove that

(i) the conjunction of $\beta(b) \sqsubseteq \beta(c)$ and $c \in I$ implies $b \in I$ for every $b, c : B$;

(ii) for every $b \in I$, there exists $c \in I$ such that $\beta(b) \ll \beta(c)$.

Note that (ii) is just saying that I is a rounded ideal w.r.t. $\ll_\beta$, so this holds. For (i), suppose that $\beta(b) \sqsubseteq \beta(c)$ and $c \in I$. By roundedness of I, there exists $c' \in I$ such that $c \ll_\beta c'$. But then $\beta(b) \sqsubseteq \beta(c) \ll \beta(c')$, so that $b \ll_\beta c'$ which implies that $b \in I$, because ideals are lower sets. □

4.11.2 Ideal completion with respect to the order relation

Lemma 4.11.7. *If $\beta : B \to D$ is a small basis for a $\mathcal{V}$-dcpo D, then $(B, \sqsubseteq_\beta)$ is an abstract $\mathcal{V}$-basis where $b \sqsubseteq_\beta c$ is defined as $\beta(b) \sqsubseteq \beta(c)$.*

Proof. The relation $\sqsubseteq_\beta$ is reflexive, so this follows from Lemma 4.10.12. □

Remark 4.11.8. The definition of an abstract $\mathcal{V}$-basis requires the relation on it to be $\mathcal{V}$-valued. Hence, for the lemma to make sense we appeal to Proposition 4.7.5 to know that D is locally small which tells us that we can substitute $\beta(b) \sqsubseteq \beta(c)$ by an equivalent type in $\mathcal{V}$.

Theorem 4.11.9. *The map $\downarrow_\beta(-) : D \to \mathcal{V}\text{-Idl}(B, \sqsubseteq_\beta)$ is the embedding in an embedding-projection pair. In particular, D is a Scott continuous retract of the algebraic dcpo $\mathcal{V}\text{-Idl}(B, \sqsubseteq_\beta)$ that has a small compact basis. Moreover, if β is a small* compact *basis, then the map is an isomorphism.*

Proof. First of all, we should check that the map is well-defined, i.e. that $\downarrow_\beta x$ is an $(B, \sqsubseteq_\beta)$-ideal. It is an inhabited subset by nullary interpolation in the basis and a semidirected lower set because the criteria of Lemma 4.11.3 are satisfied when taking $\prec$ to be $\sqsubseteq_\beta$. Secondly, the map $\downarrow_\beta(-)$ is Scott continuous by Lemma 4.11.1.
Now notice that the map $\beta : (B, \sqsubseteq_\beta) \to D$ is monotone and that the continuous map it induces by Theorem 4.10.16 is exactly the map $\bigsqcup : \mathcal{V}\text{-Idl}(B, \sqsubseteq_\beta) \to D$ that takes an ideal I to the supremum of its associated directed family $\beta \circ \mathrm{pr}_1 : (\Sigma_{b:B}(b \in I)) \to D$. Since β is a basis for D, we know that $\bigsqcup \downarrow_\beta x = x$ for every $x : D$. So it only remains to show that $\downarrow_\beta \circ \bigsqcup$ is a deflation, for which we will use Lemma 4.11.2. So suppose that $I : \mathcal{V}\text{-Idl}(B, \sqsubseteq_\beta)$ is arbitrary. Then we only need to prove that the conjunction of $\beta(b) \sqsubseteq \beta(c)$ and $c \in I$ implies $b \in I$, but this holds, because I is a lower set with respect to $\sqsubseteq_\beta$.
Finally, assume that β is a small compact basis. We show that $\downarrow_\beta \circ \bigsqcup$ is also inflationary in this case. So let I be an arbitrary ideal. By Lemma 4.11.2 it is enough to show that for every $b \in I$, there exists $c \in I$ such that $\beta(b) \ll \beta(c)$. But by assumption, $\beta(b)$ is compact, so we can simply take c to be b. □

Combining Theorems 4.10.11 and 4.11.6 and Theorems 4.11.9 and 4.10.15, we obtain the following result.

Corollary 4.11.10.
(i) A $\mathcal{V}$-dcpo D has a small basis if and only if it is isomorphic to $\mathcal{V}$-$\mathrm{Idl}(B, \prec)$ for an abstract basis $(B, \prec)$.
(ii) A $\mathcal{V}$-dcpo D has a small compact basis if and only if it is isomorphic to $\mathcal{V}$-$\mathrm{Idl}(B, \prec)$ for an abstract basis $(B, \prec)$ where $\prec$ is reflexive.

In particular, every $\mathcal{V}$-dcpo with a small basis is isomorphic to one whose order takes values in $\mathcal{V}$ and whose carrier lives in $\mathcal{V}^+$.

4.12 Structurally continuous and algebraic bilimits

We show that bilimits are closed under structural continuity/algebraicity. For the reminder of this section, fix a directed diagram of $\mathcal{V}$-dcpos $(D_i)_{i:I}$ with embedding-projection pairs $(\varepsilon_{i,j}, \pi_{i,j})_{i \sqsubseteq j \text{ in } I}$ between them, as in Section 3.6.

Now suppose that for every $i : I$, we have $\alpha_i : J_i \to D_i$ with each $J_i : \mathcal{V}$. Then we define $J_\infty :\equiv \Sigma_{i:I} J_i$ and $\alpha_\infty : J_\infty \to D_\infty$ by $(i, j) \mapsto \varepsilon_{i,\infty}(\alpha_i(j))$, where $\varepsilon_{i,\infty}$ is as in Definition 3.6.15.

Lemma 4.12.1. *If every α_i is directed and we have $\sigma : D_\infty$ such that α_i approximates σ_i, then α_∞ is directed and approximates σ.*

Proof. Observe that α_∞ is equal to the supremum, if it exists, of the directed families $(\varepsilon_{i,\infty} \circ \alpha_i)_{i:I}$ in the ind-completion of D_∞, cf. the proof of Lemma 4.3.3. Hence, for directedness of α_∞, it suffices to prove that the family $i \mapsto \varepsilon_{i,\infty} \circ \alpha_i$ is directed with respect to cofinality. The index type I is inhabited, because we are working with a directed diagram of dcpos. For semidirectedness, we will first prove that if $i \sqsubseteq i'$, then $\varepsilon_{i,\infty} \circ \alpha_i$ is cofinal in $\varepsilon_{i',\infty} \circ \alpha_{i'}$.
So suppose that $i \sqsubseteq i'$ and $j : J_i$. As α_i approximates σ_i, we have $\alpha_i(j) \ll \sigma_i$. Because $\varepsilon_{i,i'}$ is an embedding, it preserves the way-below relation (Lemma 4.2.16), so that we get $\varepsilon_{i,i'}(\alpha_i(j)) \ll \varepsilon_{i,i'}(\sigma_i) \sqsubseteq \sigma_{i'} = \bigsqcup \alpha_{i'}$. Hence, there exists $j' : J_{i'}$ with $\varepsilon_{i,i'}(\alpha_i(j)) \sqsubseteq \alpha_{i'}(j')$ which yields $\varepsilon_{i,\infty}(\alpha_i(j)) = \varepsilon_{i',\infty}(\varepsilon_{i,i'}(\alpha_i(j))) \sqsubseteq \varepsilon_{i',\infty}(\alpha_{i'}(j'))$, completing the proof that $\varepsilon_{i,\infty} \circ \alpha_i$ is cofinal in $\varepsilon_{i',\infty} \circ \alpha_{i'}$.
Now to prove that the family $i \mapsto \varepsilon_{i,\infty} \circ \alpha_i$ is semidirected with respect to cofinality, suppose we have $i_1, i_2 : I$. Since I is a directed preorder, there exists $i : I$ such that $i_1, i_2 \sqsubseteq i$. But then $\varepsilon_{i_1,\infty} \circ \alpha_{i_1}$ and $\varepsilon_{i_2,\infty} \circ \alpha_{i_2}$ are both cofinal in $\varepsilon_{i,\infty} \circ \alpha_i$ by the above.
Thus, α_∞ is directed. To see that its supremum is σ, observe that

$$
\begin{aligned}
\sigma &= \textstyle\bigsqcup_{i:I} \varepsilon_{i,\infty}(\sigma_i) && \text{(by Lemma 3.6.24)} \\
&= \textstyle\bigsqcup_{i:I} \varepsilon_{i,\infty}(\bigsqcup \alpha_i) && \text{(since } \alpha_i \text{ approximates } \sigma_i\text{)} \\
&= \textstyle\bigsqcup_{i:I} \bigsqcup \varepsilon_{i,\infty} \circ \alpha_i && \text{(by Scott continuity of } \varepsilon_{i,\infty}\text{)} \\
&= \textstyle\bigsqcup_{(i,j):J_\infty} \alpha_\infty(i, j),
\end{aligned}
$$

as desired.
Finally, we wish to show that $\alpha_\infty(i, j) \ll \sigma$ for every $(i, j) : J_\infty$. But $\varepsilon_{i,\infty}$ is an embed-

ding and therefore preserves the way-below relation while $\alpha_i(j)$ approximates σ_i, so we get $\alpha_\infty(i,j) \equiv \varepsilon_{i,\infty}(\alpha_i(j)) \ll \varepsilon_{i,\infty}(\sigma_i) \sqsubseteq \sigma$ where the final inequality holds because $\varepsilon_{i,\infty} \circ \pi_{i,\infty}$ is a deflation. $\square$

Lemma 4.12.2. *If $\alpha_i(j)$ is compact for every $i : I$ and $j : J_i$, then all the values of α_∞ are compact too.*

Proof. Let $(i,j) : J_\infty$ be arbitrary. Since $\varepsilon_{i,\infty}$ is an embedding it preserves compact elements, so $\alpha_\infty(i,j) \equiv \varepsilon_{i,\infty}(\alpha_i(j))$ is compact. $\square$

Theorem 4.12.3. *If each D_i is structurally continuous, then so is D_∞. Furthermore, if each D_i is structurally algebraic, then so is D_∞.*

Proof. Let $\sigma : D_\infty$ be arbitrary. By structural continuity of each D_i, we have a directed family $\alpha_i : J_i \to D_i$ approximating σ_i. Hence, by Lemma 4.12.1, the family α_∞ is directed and approximates σ, proving the structural continuity of D_∞. Now if each D_i is structurally algebraic, then D_∞ is structurally algebraic by Lemma 4.12.2 and the above. $\square$

Note that we do not expect to be able to prove that D_∞ is continuous if each D_i is, because it would require an instance of the axiom of choice to get the continuity structures on each D_i and without those we have nothing to operate on.

Theorem 4.12.4. *If each D_i has a small basis $\beta_i : B_i \to D_i$, then the map β_∞ defined by $\beta_\infty : (B_\infty :\equiv (\Sigma_{i:I} B_i)) \xrightarrow{(i,b)\mapsto \varepsilon_{i,\infty}(\beta_i(b))} D_\infty$ is a small basis for D_∞. Furthermore, if each β_i is a small compact basis, then β_∞ is a small compact basis too.*

Proof. First of all, we must show that $\beta_\infty(i,b) \ll \sigma$ is small for every $i : I$, $b : B_i$ and $\sigma : D_\infty$. We claim that this is the case as the way-below relation on D_∞ has small values. By Proposition 4.4.13 and Theorem 4.12.3, it suffices to prove that D_∞ is locally small. But this holds by Proposition 3.6.29 as each D_i is locally small by Proposition 4.7.5.
It remains to prove that, for an arbitrary element $\sigma : D_\infty$, the family $\downarrow_{\beta_\infty} \sigma$ given by $\left(\Sigma_{(i,b):B_\infty} \beta_\infty(i,b) \ll \sigma\right) \xrightarrow{\beta_\infty \circ \mathrm{pr}_1} D_\infty$ is directed with supremum σ. Note that for every $i : I$ and $b : B_i$, we have that $\beta_i(b) \ll \sigma_i$ implies

$$\beta_\infty(i,b) \equiv \varepsilon_{i,\infty}(\beta_i(b)) \ll \varepsilon_{i,\infty}(\sigma_i) \sqsubseteq \sigma,$$

since Lemma 4.2.16 tells us that the embedding $\varepsilon_{i,\infty}$ preserves the way-below relation. Hence, the identity induces a well-defined map

$$\iota : \left(\Sigma_{i:I}\Sigma_{b:B_i} \beta_i(b) \ll \sigma_i\right) \to \left(\Sigma_{(i,b):B_\infty} \beta_\infty(i,b) \ll \sigma\right).$$

Lemma 4.7.9 now tells us that we only need to show that $\downarrow_{\beta_\infty} \sigma \circ \iota$ is directed and has supremum σ. But if we write $\alpha_i : \left(\Sigma_{b:B_i} \beta_i(b) \ll \sigma_i\right) \to D_i$ for the map $b \mapsto \beta_i(b)$, then we see that $\downarrow_{\beta_\infty} \sigma \circ \iota$ is given by α_∞, as defined at the start of this section. But

then α_∞ is indeed seen to be directed with supremum σ by virtue of Lemma 4.12.1 and the fact that α_i approximates σ_i.
Finally, if every β_i is a small compact basis, then β_∞ is also a small compact basis because by Lemma 4.8.4 all we need to know is that $\beta_\infty(i,b) \equiv \varepsilon_{i,\infty}(\beta_i(b))$ is compact for every $i : I$ and $b : B_i$. But this follows from the fact that embeddings preserve compactness and that each $\beta_i(b)$ is compact. □

4.13 Exponentials with small (compact) bases

Just as in the classical, impredicative setting, the exponential of two continuous dcpos need not be continuous [Jun89]. However, with some work, we are able to show that E^D has a small basis provided that both D and E do and that E has all (not necessarily directed) $\mathcal{V}$-suprema. We first establish this for small compact bases using *step functions* and then derive the result for compact bases using Theorem 4.11.9.

4.13.1 Single step functions

Suppose that we have a dcpo D and a pointed dcpo E. Classically [GHK+03, Exercise II-2.31], the single step function given by $d : D$ and $e : E$ is defined as

$$\begin{aligned} (\!| d \Rightarrow e |\!) &: D \to E \\ x &\mapsto \begin{cases} e & \text{if } d \sqsubseteq x; \\ \bot & \text{otherwise.} \end{cases} \end{aligned}$$

Constructively, we can't expect to make this case distinction, so we define single step functions using subsingleton suprema instead.

Definition 4.13.1 (Single step function, $(\!| d \Rightarrow e |\!)$)**.** The *single step function* given by two elements $d : D$ and $e : E$, where D is a locally small $\mathcal{V}$-dcpo and E is a pointed $\mathcal{V}$-dcpo, is the function $(\!| d \Rightarrow e |\!) : D \to E$ given by mapping $x : D$ to the supremum of the family indexed by the subsingleton $d \sqsubseteq x$ that is constantly e.

Remark 4.13.2. Recall from Lemma 3.3.10 that the supremum of a subsingleton-indexed family $\alpha : P \to E$ is given by the supremum of the directed family $\mathbf{1} + P \to E$ defined by $\mathrm{inl}(\star) \mapsto \bot$ and $\mathrm{inr}(p) \mapsto \alpha(p)$. Note that we need D to be locally small, because we need $d \sqsubseteq x$ to be a subsingleton in $\mathcal{V}$ to use the $\mathcal{V}$-directed-completeness of E.

Lemma 4.13.3. *If $d : D$ is compact, then $(\!| d \Rightarrow e |\!)$ is Scott continuous for every $e : E$.*

Proof. Suppose that $d : D$ is compact and that $\alpha : I \to D$ is a directed family. We first show that $(\!| d \Rightarrow e |\!)(\bigsqcup \alpha)$ is an upper bound of $(\!| d \Rightarrow e |\!) \circ \alpha$. So let $i : I$ be arbitrary. Then we have to prove $\bigsqcup_{d \sqsubseteq \alpha_i} e \sqsubseteq \bigsqcup (\!| d \Rightarrow e |\!) \circ \alpha$. Since the supremum gives a lower bound of upper bounds, it suffices to prove that $e \sqsubseteq \bigsqcup (\!| d \Rightarrow e |\!) \circ \alpha$ whenever $d \sqsubseteq \alpha_i$.

But in this case we have $e = (\!|d \Rightarrow e|\!)(\alpha_i) \sqsubseteq \bigsqcup (\!|d \Rightarrow e|\!) \circ \alpha$, so we are done.
To see that $(\!|d \Rightarrow e|\!)(\bigsqcup \alpha)$ is a lower bound of upper bounds, suppose that we are given $y : E$ such that y is an upper bound of $(\!|d \Rightarrow e|\!) \circ \alpha$. We are to prove that $(\bigsqcup_{d \sqsubseteq \bigsqcup \alpha} e) \sqsubseteq y$. Note that it suffices for $d \sqsubseteq \bigsqcup \alpha$ to imply $e \sqsubseteq y$. So assume that $d \sqsubseteq \bigsqcup \alpha$. By compactness of d there exists $i : I$ such that $d \sqsubseteq \alpha_i$ already. But then $e = (\!|d \Rightarrow e|\!)(\alpha_i) \sqsubseteq y$, as desired. ☐

Lemma 4.13.4. *A Scott continuous function $f : D \to E$ is above the single step function $(\!|d \Rightarrow e|\!)$ with $d : D$ compact if and only if $e \sqsubseteq f(d)$.*

Proof. Suppose that $(\!|d \Rightarrow e|\!) \sqsubseteq f$. Then $(\!|d \Rightarrow e|\!)(d) = e \sqsubseteq f(d)$, proving one implication. Now assume that $e \sqsubseteq f(d)$ and let $x : D$ be arbitrary. To prove that $(\!|d \Rightarrow e|\!)(x) \sqsubseteq f(x)$, it suffices that $e \sqsubseteq f(x)$ whenever $d \sqsubseteq x$. But if $d \sqsubseteq x$, then $e \sqsubseteq f(d) \sqsubseteq f(x)$ by monotonicity of f. ☐

Lemma 4.13.5. *If d and e are compact, then so is $(\!|d \Rightarrow e|\!)$ in the exponential E^D.*

Proof. Suppose that we have a directed family $\alpha : I \to E^D$ such that $(\!|d \Rightarrow e|\!) \sqsubseteq \bigsqcup \alpha$. Then we consider the directed family $\alpha^d : I \to E$ given by $i \mapsto \alpha_i(d)$. We claim that $e \sqsubseteq \bigsqcup \alpha^d$. Indeed, by Lemma 4.13.4 and our assumption that $(\!|d \Rightarrow e|\!) \sqsubseteq \bigsqcup \alpha$ we get $e \sqsubseteq (\bigsqcup \alpha)(d) = \bigsqcup \alpha^d$. Now by compactness of e, there exists $i : I$ such that $e \sqsubseteq \alpha^d(i) \equiv \alpha_i(d)$ already. But this implies $(\!|d \Rightarrow e|\!) \sqsubseteq \alpha_i$ by Lemma 4.13.4 again, finishing the proof. ☐

4.13.2 Exponentials with small compact bases

Fix $\mathcal{V}$-dcpos D and E with small compact bases $\beta_D : B_D \to D$ and $\beta_E : B_E \to D$ and moreover assume that E has suprema for all (not necessarily directed) families indexed by types in $\mathcal{V}$. We are going to construct a small compact basis on the exponential E^D.

Lemma 4.13.6. *If E is sup-complete, then every Scott continuous function $f : D \to E$ is the supremum of the collection of single step functions $\left((\!|\beta_D(b) \Rightarrow \beta_E(c)|\!)\right)_{b:B_D,c:B_E}$ that are below f.*

Proof. Note that f is an upper bound by definition, so it remains to prove that it is the least. Therefore suppose we are given an upper bound $g : D \to E$. We have to prove that $f(x) \sqsubseteq g(x)$ for every $x : D$, so let $x : D$ be arbitrary. Now $x = \bigsqcup \downarrow_{\beta_D} x$, because β_D is a small compact basis for D, so by Scott continuity of f and g, it suffices to prove that $f(\beta_D(b)) \sqsubseteq g(\beta_D(b))$ for every $b : B_D$. So let $b : B_D$ be arbitrary. Since β_E is a small compact basis for E, we have $f(\beta_D(b)) = \bigsqcup \downarrow_{\beta_E} f(\beta_D(b))$. So to prove $f(\beta_D(b)) \sqsubseteq g(\beta_D(b))$ it is enough to know that $\beta_E(c) \sqsubseteq g(\beta_D(b))$ for every $c : B_E$ with $\beta_E(c) \sqsubseteq f(\beta_D(b))$. But for such $c : B_E$ we have $(\!|\beta_D(b) \Rightarrow \beta_E(c)|\!) \sqsubseteq f$ and therefore $(\!|\beta_D(b) \Rightarrow \beta_E(c)|\!) \sqsubseteq g$ because f is an upper bound of such single step functions, and hence $\beta_E(c) \sqsubseteq g(\beta_D(b))$ by Lemma 4.13.4, as desired. ☐

Definition 4.13.7 (Directification)**.** In a $\mathcal{V}$-sup-complete poset P, the *directification* of a family $\alpha : I \to P$ is the family $\bar{\alpha} : \mathrm{List}(I) \to P$ inductively defined by $[] \mapsto \bot$ and $i :: l \mapsto \alpha_i \vee \bar{\alpha}(l)$, where $\bot$ denotes the least element of P and $\vee$ the binary join. It is clear that $\bar{\alpha}$ has the same supremum as α, and by concatenating lists, one sees that the directification yields a directed family, hence the name.

Lemma 4.13.8. *If each element of a family into a sup-complete dcpo is compact, then so are all elements of its directification.*

Proof. By induction, Example 4.2.4 and Lemma 4.2.7. □

Let us write $\sigma : B_D \times B_E \to E^D$ for the map that takes (b, c) to the single step function $(\!|\beta_D(b) \Rightarrow \beta_E(c)|\!)$ and $\beta : B :\equiv \mathrm{List}(B_D \times B_E) \to E^D$ for its directification, which exists because E^D is $\mathcal{V}$-sup-complete as E is and suprema are calculated pointwise.

Theorem 4.13.9. *The map β is a small compact basis for the exponential E^D, where E is assumed to be sup-complete.*

Proof. Firstly, every element in the image of β is compact by Lemmas 4.13.5 and 4.13.8. Secondly, for every $b : B$ and Scott continuous map $f : D \to E$, the type $\beta(b) \sqsubseteq f$ is small, because E^D is locally small by Proposition 4.7.11. Thirdly, for every such f, the family $(\Sigma_{b:B}(\beta(b) \sqsubseteq f)) \xrightarrow{\beta \circ \mathrm{pr}_1} E^D$ is directed because β is the directification of σ. Finally, this family has supremum f because of Lemma 4.13.6. □

4.13.3 Exponentials with small bases

We now present a variation of Theorem 4.13.9 but for (sup-complete) dcpos with small bases. In fact, we will prove it using Theorem 4.13.9 and the theory of Scott continuous retracts (Theorem 4.11.9 in particular).

Definition 4.13.10 (Closure under finite joins)**.** A small basis $\beta : B \to D$ for a sup-complete poset is *closed under finite joins* if we have $b_\bot : B$ with $\beta(b_\bot) = \bot$ and a map $\vee : B \to B \to B$ such that $\beta(b \vee c) = \beta(b) \vee \beta(c)$ for every $b, c : B$.

Lemma 4.13.11. *If D is a sup-complete dcpo with a small basis $\beta : B \to D$, then the directification of β is also a small basis for D. Moreover, by construction, it is closed under finite joins.*

Proof. Since β is a small basis for D, it follows by Proposition 4.7.5 that the way-below relation on D is small-valued. Hence, writing $\bar{\beta}$ for the directification of β, it remains to prove that $\downarrow_{\bar{\beta}} x$ is directed with supremum x for every $x : D$. But this follows easily from Lemma 4.7.9, because $\downarrow_{\beta} x$ is directed with supremum x and this family is equal to the composite $(\Sigma_{b:B}(\beta(b) \ll x)) \xhookrightarrow{b \mapsto [b]} (\Sigma_{l:\mathrm{List}(B)}(\bar{\beta}(l) \ll x)) \xrightarrow{\bar{\beta} \circ \mathrm{pr}_1} D$. □

Lemma 4.13.12. *If D is a $\mathcal{V}$-sup-complete poset with a small basis $\beta : B \to D$ that is closed under finite joins, then the ideal-completion $\mathcal{V}\text{-Idl}(B, \sqsubseteq)$ is $\mathcal{V}$-sup-complete too.*

Proof. Since the $\mathcal{V}$-ideal completion is $\mathcal{V}$-directed complete, it suffices to show that $\mathcal{V}\text{-Idl}(B, \sqsubseteq)$ has finite joins. Since $\beta : B \to D$ is closed under finite joins, we have $b_\perp : B$ with $\beta(b_\perp) = \perp$ and we easily see that $\{b_\perp\}$ is the least element of $\mathcal{V}\text{-Idl}(B, \sqsubseteq)$. Now suppose that we have two ideals $I, J : \mathcal{V}\text{-Idl}(B, \sqsubseteq)$ and consider the subset

$$K :\equiv \{b \in B \mid \exists_{b_0,\dots,b_{n-1} \in I} \exists_{c_0,\dots,c_{m-1} \in J} (\beta(b) \sqsubseteq \beta(b_0 \vee \cdots \vee b_{n-1} \vee c_0 \vee \cdots \vee c_{m-1}))\}.$$

Observe that K is a lower set and that it is directed as B is closed under finite joins. Thus, $K \in \mathcal{V}\text{-Idl}(B, \sqsubseteq)$. We claim that K is the join of I and J. First of all, I and J are both subsets of K, so it remains to prove that K is the least upper bound. To this end, suppose that we have an ideal L that includes I and J, and let $b \in K$ be arbitrary. We show that $b \in L$. Since $b \in K$, there exist $b_0, \dots, b_{n-1} \in I$ and $c_0, \dots, c_{m-1} \in J$ such that $\beta(b) \sqsubseteq \beta(b_0 \vee \cdots \vee b_{n-1} \vee c_0 \vee \cdots \vee c_{m-1})$. Then $b_0, \dots, b_{n-1} \in L$ and there exists $b \in L$ such that $\beta(b_0), \dots, \beta(b_{n-1}) \sqsubseteq \beta(b)$ as L is directed. But L is a lower set, so L must also contain $\beta(b_0 \vee \cdots \vee b_{n-1})$. Similarly, L contains $\beta(c_0 \vee \cdots \vee c_{m-1})$. Finally, using a similar argument once again, we get that L contains $\beta(b_0 \vee \cdots \vee b_{n-1} \vee c_0 \vee \cdots \vee c_{m-1})$. But $\beta(b)$ is below this element, so L must contain it, finishing the argument that $K \subseteq L$. Hence, K is the least upper bound of I and J, completing the proof. $\square$

Theorem 4.13.13. *The exponential E^D of dcpos has a specified small basis if D and E have specified small bases and E is sup-complete.*

Proof. Suppose that $\beta_D : B_D \to D$ and $\beta_E : B_E \to E$ are small bases and that E is sup-complete. By Lemma 4.13.11 we can assume that $\beta_E : B_E \to E$ is closed under finite joins. We will write D' and E' for the respective ideal completions $\mathcal{V}\text{-Idl}\,(B_D, \sqsubseteq)$ and $\mathcal{V}\text{-Idl}(B_E, \sqsubseteq)$. Then Theorem 4.11.9 tells us that we have Scott continuous retracts $D \underset{r_D}{\overset{s_D}{\rightleftarrows}} D'$ and $E \underset{r_E}{\overset{s_E}{\rightleftarrows}} E'$. Composition yields a Scott continuous retract $E^D \underset{r}{\overset{s}{\rightleftarrows}} E'^{D'}$ where $s(f) :\equiv s_E \circ f \circ r_D$ and $r(g) :\equiv r_E \circ g \circ s_D$. Now D' and E' have small compact basis by Theorem 4.10.15 and E' is sup-complete by Lemma 4.13.12. Therefore, $E'^{D'}$ has a small basis by Theorem 4.13.9. Finally, Theorem 4.7.10 tells us that the retraction r yields a small basis on E^D, as desired. $\square$

Note how, unlike Theorem 4.13.9, the above theorem does not give an explicit description of the small basis for the exponential. It may be possible to do so using function graphs, as is done in the classical setting of effective domain theory in [Smy77, Section 4.1].

4.14 Notes

This chapter is largely based on our work [dJE21a]. In particular, Sections 4.2 and 4.10 are expanded and revised versions of parts of that paper. Some of the ideas for the arguments in Sections 4.12 and 4.13 were already present in the expanded version of [dJE21a]. The present treatment of continuous and algebraic dcpos and small (compact) bases is significantly different from that of [dJE21a]. In the published work, our definition of continuous dcpo was an amalgamation of pseudocontinuity and having a small basis, although it did not imply local smallness. In this chapter we have disentangled the two notions and based our definition of continuity on that of [JJ82] without making any reference to a basis. The current notion of a small basis is simpler and slightly stronger than that of [dJE21a], which allows us to prove that having a small basis is equivalent to being presented by ideals.

This idea of a small basis is similar, but different to Aczel's notion of a "set-generated" dcpo [Acz06, Section 6.4] in the context of constructive set theory, and a similar smallness criterion in a categorical context also appears in [JJ82, Proposition 2.16]. While Aczel requires the set $\{b \in B \mid b \sqsubseteq x\}$ to be directed, we use the way-below relation (or compact elements) in line with the usual definition of a basis [AJ94, Section 2.2.6]. The particular case of a dcpo with a small compact basis is similar to the notion of an accessible category [MP89].

As mentioned before, our treatment of continuous dcpos is based on the work [JJ82] of Johnstone and Joyal, but we use the propositional truncation to ensure that the type of continuous dcpos is a subtype of the type of dcpos. Moreover, our discussion of pseudocontinuity (Section 4.5) is new.

Abstract bases were introduced by Smyth [Smy77] under the name "R-structures", but our treatment of them and the rounded ideal completion is closer to that of [AJ94, Section 2.2.6], although ours is based on families and avoids impredicative, set-theoretic constructions.

Finally, the example of the ideal completion of the dyadics in Section 4.10.2 and Theorem 4.13.13 were suggested to me by Martín Escardó.

Chapter 5

Applications in semantics of programming languages

We present two applications of domain theory in the semantics of programming languages. The first application is Scott's [Sco72] famous D_∞: a construction of a nontrivial pointed dcpo D that is isomorphic to its self-exponential D^D. This allows one to view functions as elements (and vice versa) and hence to give a genuine model of the untyped λ-calculus where self-application is fundamental. The construction works by instantiating the bilimit machinery from Section 3.6 with a particular diagram. It is noteworthy that this construction is possible in our predicative setting given that Scott's D_∞ is obtained by iterating exponentials where it is not obvious that this does not lead to ever-increasing universes. While the applications can be developed fully using the basic theory of dcpos (as set out in Chapter 3) the exposition in Chapter 4 allows us to conclude that Scott's D_∞ is algebraic. In fact, it has a small compact basis.

The second application is a constructive and predicative account of the Scott [Sco93] model of the typed programming language PCF [Plo77], including the fundamental soundness and computational adequacy theorems, formulated and proved originally by Plotkin [Plo77]. The lifting from Section 3.4 is the essential ingredient in our constructive treatment and the model also illustrates the usefulness of the least fixed point theorem (Theorem 3.5.9). We employ computational adequacy in our investigations into semidecidability and this is also where the theory on indexed W-types (Section 2.12) will find application.

5.1 Scott's D_∞ model of the untyped λ-calculus

We construct Scott's D_∞ [Sco72] predicatively. Formulated precisely, we construct a pointed $D_\infty : \mathcal{U}_0\text{-DCPO}_{\mathcal{U}_1,\mathcal{U}_1}$ such that D_∞ is isomorphic to its self-exponential $D_\infty^{D_\infty}$, employing the machinery from Section 3.6.

Definition 5.1.1 (D_n)**.** We inductively define pointed dcpos $D_n : \mathcal{U}_0\text{-DCPO}_{\mathcal{U}_1,\mathcal{U}_1}$ for every natural number n by setting $D_0 :\equiv \mathcal{L}_{\mathcal{U}_0}(\mathbf{1}_{\mathcal{U}_0})$ and $D_{n+1} :\equiv D_n^{D_n}$.

In light of Remark 3.2.10 we highlight the fact that every D_n is a $\mathcal{U}_0$-dcpo with carrier in $\mathcal{U}_1$ by the discussion of universe parameters of exponentials in Remark 3.5.7.

Definition 5.1.2 (ε_n, π_n)**.** We inductively define for every natural number n, two Scott continuous maps $\varepsilon_n : D_n \to D_{n+1}$ and $\pi_n : D_{n+1} \to D_n$:

(i)
- $\varepsilon_0 : D_0 \to D_1$ is given by mapping $x : D_0$ to the continuous function that is constantly x,
- $\pi_0 : D_1 \to D_0$ is given by evaluating a continuous function $f : D_0 \to D_0$ at $\perp$ which is itself continuous by Proposition 3.5.8,

(ii)
- $\varepsilon_{n+1} : D_{n+1} \to D_{n+2}$ takes a continuous function $f : D_n \to D_n$ to the continuous composite $D_{n+1} \xrightarrow{\pi_n} D_n \xrightarrow{f} D_n \xrightarrow{\varepsilon_n} D_{n+1}$, and
- $\pi_{n+1} : D_{n+2} \to D_{n+1}$ takes a continuous function $f : D_{n+1} \to D_{n+1}$ to the continuous composite $D_n \xrightarrow{\varepsilon_n} D_{n+1} \xrightarrow{f} D_{n+1} \xrightarrow{\pi_n} D_n$.

Lemma 5.1.3. *The maps (ε_n, π_n) form an embedding-projection pair for every natural number n.*

Proof. We prove this by induction on n. For $n \equiv 0$ and arbitrary $x : D_0$, we have

$$\pi_0(\varepsilon_0(x)) \equiv \pi_0(\mathrm{const}_x) \equiv \mathrm{const}_x(\perp) \equiv x,$$

so ε_0 is indeed a section of π_0. Moreover, for arbitrary $f : D_1$, we have

$$\varepsilon_0(\pi_0(f)) \equiv \varepsilon_0(f(\perp)) \equiv \mathrm{const}_{f(\perp)},$$

so that for arbitrary $x : D_0$ we get $(\varepsilon_0(\pi_0(f)))(x) \equiv f(\perp) \sqsubseteq f(x)$ by monotonicity of f, proving that $\varepsilon_0 \circ \pi_0$ is deflationary.
Now suppose that the result holds for a natural number n; we prove it for $n+1$. For arbitrary $f : D_n \to D_n$, we calculate that

$$\pi_{n+1}(\varepsilon_{n+1}(f)) \equiv \pi_n \circ \varepsilon_{n+1}(f) \circ \varepsilon_n \equiv \pi_n \circ \varepsilon_n \circ f \circ \pi_n \circ \varepsilon_n = f,$$

as ε_n is a section of π_n by induction hypothesis. The proof that $\varepsilon_{n+1} \circ \pi_{n+1}$ is a deflation is similar. □

In order to apply the tools from Section 3.6, we will need embedding-projection pairs $(\varepsilon_{n,m}, \pi_{n,m})$ from D_n to D_m whenever $n \leq m$.

Definition 5.1.4 ($\varepsilon_{n,m}$, $\pi_{n,m}$)**.** We define Scott continuous maps $\varepsilon_{n,m} : D_n \to D_m$ and $\pi_{n,m} : D_m \to D_n$ for every two natural numbers $n \leq m$ as follows:

(i) $\varepsilon_{n,m}$ and $\pi_{n,m}$ are both defined to be the identity if $n = m$;
(ii) if $n < m$, then we define $\varepsilon_{n,m}$ as the composite

$$D_n \xrightarrow{\varepsilon_n} D_{n+1} \to \cdots \to D_{m-1} \xrightarrow{\varepsilon_{m-1}} D_m$$

and $\pi_{n,m}$ as the composite

$$D_m \xrightarrow{\pi_m} D_{m-1} \longrightarrow \cdots \longrightarrow D_{n+1} \xrightarrow{\pi_n} D_n$$

which yields embedding-projection pairs as they are compositions of them.

Instantiating the framework of Section 3.6 with the above diagram of objects $D_n : \mathcal{U}_0\text{-DCPO}_{\mathcal{U}_1,\mathcal{U}_1}$, we arrive at the construction of D_∞ and appropriate embedding-projection pairs. Observe that D_∞ is a $\mathcal{U}_0$-dcpo with carrier and order taking values in $\mathcal{U}_1$, just like each D_n, as was also mentioned in Remark 3.6.8.

Definition 5.1.5 (D_∞)**.** Applying Definitions 3.6.6, 3.6.9 and 3.6.15 to the above diagram yields $D_\infty : \mathcal{U}_0\text{-DCPO}_{\mathcal{U}_1,\mathcal{U}_1}$ with embedding-projection pairs $(\varepsilon_{n,\infty}, \pi_{n,\infty})$ from D_n to D_∞ for every natural number n.

Lemma 5.1.6. *The function $\pi_n : D_{n+1} \longrightarrow D_n$ is strict for every natural number n. Hence, so is $\pi_{n,m}$ whenever $n \leq m$.*

Proof. Both statements are proved by induction. □

Proposition 5.1.7. *The dcpo D_∞ is pointed.*

Proof. Since every D_n is pointed, we can consider the function $\sigma : \Pi_{n:\mathbb{N}} D_n$ given by $\sigma(n) :\equiv \perp_{D_n}$. Then σ is an element of D_∞ by Lemma 5.1.6 and it is the least, so D_∞ is indeed pointed. □

We now work towards showing that D_∞ is isomorphic to the exponential $D_\infty^{D_\infty}$. Note that this exponential is again an element of $\mathcal{U}_0\text{-DCPO}_{\mathcal{U}_1,\mathcal{U}_1}$ by Remark 3.5.7, so the universe parameters do not increase.

Definition 5.1.8 (Φ_n)**.** For every natural number n, we define the continuous maps

$$\begin{aligned} \Phi_{n+1} : D_{n+1} &\longrightarrow D_\infty^{D_\infty} \\ f &\mapsto (D_\infty \xrightarrow{\pi_{n,\infty}} D_n \xrightarrow{f} D_n \xrightarrow{\varepsilon_{n,\infty}} D_\infty) \end{aligned}$$

and $\Phi_0 : D_0 \longrightarrow D_\infty^{D_\infty}$ as $\Phi_1 \circ \varepsilon_0$.

Lemma 5.1.9. *For every two natural numbers $n \leq m$, the diagram*

$$\begin{tikzcd} D_n \arrow[rr, "\Phi_n"] \arrow[dr, "\varepsilon_{n,m}"'] & & D_\infty^{D_\infty} \\ & D_m \arrow[ur, "\Phi_m"'] & \end{tikzcd}$$

commutes.

Proof. By induction on the difference of the two natural numbers, it suffices to prove that for every natural number n, the diagram

$$\begin{tikzcd} D_n \arrow[rr, "\Phi_n"] \arrow[dr, "\varepsilon_n"'] & & D_\infty^{D_\infty} \\ & D_{n+1} \arrow[ur, "\Phi_{n+1}"'] & \end{tikzcd}$$

commutes. But this follows from Lemma 3.6.19 and unfolding the definition of Φ_n. □

Definition 5.1.10 (Φ)**.** The map $\Phi : D_\infty \to D_\infty^{D_\infty}$ is defined as the unique Scott continuous map induced by the Φ_n via Theorem 3.6.26.

Lemma 5.1.11. *For $\sigma : D_\infty$ we have $\Phi(\sigma) = \bigsqcup_{n:\mathbb{N}} \Phi_{n+1}(\sigma_{n+1})$.*

Proof. Recalling the proof of Theorem 3.6.26 we have $\Phi(\sigma) \equiv \bigsqcup_{n:\mathbb{N}} \Phi_n(\sigma_n)$, from which the claim follows easily. □

We now define a map in the other direction using that D_∞ is also the limit.

Definition 5.1.12 (Ψ_n)**.** For every natural number n, we define the continuous maps

$$\begin{aligned} \Psi_{n+1} : D_\infty^{D_\infty} &\to D_{n+1} \\ f &\mapsto (D_n \xrightarrow{\varepsilon_{n,\infty}} D_\infty \xrightarrow{f} D_\infty \xrightarrow{\pi_{n,\infty}} D_n) \end{aligned}$$

and $\Psi_0 : D_\infty^{D_\infty} \to D_0$ as $\pi_0 \circ \Psi_1$.

Lemma 5.1.13. *For every two natural numbers $n \leq m$, the diagram*

$$\begin{tikzcd} D_\infty^{D_\infty} \arrow[rr, "\Psi_n"] \arrow[dr, "\Psi_m"'] & & D_n \\ & D_m \arrow[ur, "\pi_{n,m}"'] & \end{tikzcd}$$

commutes.

Proof. Similar to Lemma 5.1.9. □

Definition 5.1.14 (Ψ)**.** The map $\Psi : D_\infty^{D_\infty} \to D_\infty$ is defined as the unique Scott continuous map induced by the Ψ_n via Theorem 3.6.20.

Lemma 5.1.15. *For $f : D_\infty^{D_\infty}$ we have $\Psi(f) = \bigsqcup_{n:\mathbb{N}} \varepsilon_{n+1,\infty}(\Psi_{n+1}(f))$.*

Proof. Notice that

$$\begin{aligned}
\Psi(f) &= \bigsqcup_{n:\mathbb{N}} \varepsilon_{n,\infty}\big(\pi_{n,\infty}(\Psi(f))\big) && \text{(by Lemma 3.6.24)}\\
&= \bigsqcup_{n:\mathbb{N}} \varepsilon_{n,\infty}(\Psi_n(f)) && \text{(by Equation 3.6.22)},
\end{aligned}$$

from which the claim follows easily. □

Theorem 5.1.16. *The maps Φ and Ψ are inverses and hence, D_∞ is isomorphic to $D_\infty^{D_\infty}$.*

Proof. For arbitrary $\sigma : D_\infty$ we calculate that

$$\begin{aligned}
\Psi(\Phi(\sigma)) &= \Psi\big(\bigsqcup_{n:\mathbb{N}} \Phi_{n+1}(\sigma_{n+1})\big) && \text{(by Lemma 5.1.11)}\\
&= \bigsqcup_{n:\mathbb{N}} \Psi(\Phi_{n+1}(\sigma_{n+1})) && \text{(by Scott continuity of } \Psi)\\
&= \bigsqcup_{n:\mathbb{N}} \bigsqcup_{m:\mathbb{N}} \varepsilon_{m+1,\infty}(\Psi_{m+1}(\Phi_{n+1}(\sigma_{n+1}))) && \text{(by Lemma 5.1.15)}\\
&= \bigsqcup_{n:\mathbb{N}} \varepsilon_{n+1,\infty}(\Psi_{n+1}(\Phi_{n+1}(\sigma_{n+1})))\\
&\equiv \bigsqcup_{n:\mathbb{N}} \varepsilon_{n+1,\infty}\big(\pi_{n,\infty} \circ \varepsilon_{n,\infty} \circ \sigma_{n+1} \circ \pi_{n,\infty} \circ \varepsilon_{n,\infty}\big) && \text{(by definition)}\\
&= \bigsqcup_{n:\mathbb{N}} \varepsilon_{n+1,\infty}(\sigma_{n+1}) && \text{(since } \pi_{n,\infty} \circ \varepsilon_{n,\infty} = \mathrm{id})\\
&= \sigma && \text{(by Lemma 3.6.24)},
\end{aligned}$$

so Φ is indeed a section of Ψ. Moreover, for arbitrary $f : D_\infty^{D_\infty}$ we calculate that

$$\begin{aligned}
\Phi(\Psi(f)) &= \Phi\big(\bigsqcup_{n:\mathbb{N}} \varepsilon_{n+1,\infty}(\Psi_{n+1}(f))\big) && \text{(by Lemma 5.1.15)}\\
&= \bigsqcup_{n:\mathbb{N}} \Phi\big(\varepsilon_{n+1,\infty}(\Psi_{n+1}(f))\big) && \text{(by Scott continuity of } \Phi)\\
&= \bigsqcup_{n:\mathbb{N}} \bigsqcup_{m:\mathbb{N}} \Phi_{m+1}\big(\pi_{m+1,\infty}\big(\varepsilon_{n+1,\infty}(\Psi_{n+1}(f))\big)\big) && \text{(by Lemma 5.1.11)}\\
&= \bigsqcup_{n:\mathbb{N}} \Phi_{n+1}\big(\pi_{n+1,\infty}\big(\varepsilon_{n+1,\infty}(\Psi_{n+1}(f))\big)\big)\\
&= \bigsqcup_{n:\mathbb{N}} \Phi_{n+1}(\Psi_{n+1}(f)) && \text{(since } \pi_{n+1,\infty} \circ \varepsilon_{n+1,\infty} = \mathrm{id})\\
&\equiv \bigsqcup_{n:\mathbb{N}} \big(\varepsilon_{n,\infty} \circ \pi_{n,\infty} \circ f \circ \varepsilon_{n,\infty} \circ \pi_{n,\infty}\big) && \text{(by definition)}\\
&= \big(\bigsqcup_{n:\mathbb{N}} \varepsilon_{n,\infty} \circ \pi_{n,\infty}\big) \circ f \circ \big(\bigsqcup_{m:\mathbb{N}} \varepsilon_{m,\infty} \circ \pi_{m,\infty}\big)\\
&= f && \text{(by Lemma 3.6.25)},
\end{aligned}$$

finishing the proof. □

Remark 5.1.17. Of course, Theorem 5.1.16 is only interesting when D_∞ is not the trivial one-element dcpo. Fortunately, D_∞ has (infinitely) many elements besides the least element$\perp_{D_\infty}$. For instance, we can consider $x :\equiv \eta(\star) : D_0$ and observe that $\varepsilon_{0,\infty}(x)$ is an element of D_∞ not equal to $\perp_{D_\infty}$, since $x \neq \perp_{D_0}$.

Theorem 5.1.18. *Scott's D_∞ has a small compact basis and in particular is (structurally) algebraic.*

Proof. By Example 4.9.2 the $\mathcal{U}_0$-dcpo D_0 has a small compact basis. Moreover, it is not just a $\mathcal{U}_0$-dcpo as it has suprema for all (not necessarily directed) families indexed by types in $\mathcal{U}_0$, as D_0 is isomorphic to $\Omega_{\mathcal{U}_0}$. Hence, by induction it follows that each D_n is $\mathcal{U}_0$-sup-complete. Therefore, by induction and Theorem 4.13.9 we get a small compact basis for each D_n. Thus, by Theorem 4.12.4, the bilimit D_∞ has a small basis too. □

5.2 Scott's model of the programming language PCF

PCF [Plo77] is a typed λ-calculus with a base type for natural numbers and additional constants. The full syntax of PCF and its reduction rules (operational semantics) are described in Section 5.2.1. For example, we have numerals $\underline{n}$ of the base type ι corresponding to natural numbers and basic operations on them, such as a predecessor term `pred` and a term `ifz` that allows us to perform case distinction on whether an input is zero or not. The most striking feature of PCF is its fixed point combinator $\mathtt{fix}_\sigma$ for every PCF type σ. The idea is that for a term t of function type $\sigma \Rightarrow \sigma$, the term $\mathtt{fix}_\sigma\, t$ of type σ is a fixed point of t. The use of `fix` is that it gives us general recursion. The operational semantics of PCF is a reduction strategy that allows us to compute in PCF. We write $s \triangleright t$ for s *reduces to* t. We show a few examples below:

$$\mathtt{pred}\,\underline{0} \triangleright \underline{0}; \quad \mathtt{pred}\,\underline{n+1} \triangleright \underline{n}; \quad \mathtt{ifz}\,s\,t\,\underline{0} \triangleright s; \quad \mathtt{ifz}\,s\,t\,\underline{n+1} \triangleright t; \quad \mathtt{fix}\,t \triangleright t(\mathtt{fix}\,t).$$

We see that `pred` indeed acts as a predecessor function and that `ifz` performs case distinction on whether its third argument is zero or not. The reduction rule for `fix` reflects that $\mathtt{fix}\,t$ is a fixed point of t and may be seen as an unfolding (of a recursive definition).

Another way to give meaning to the PCF terms is through denotational semantics, that is, by giving a model of PCF. A model of PCF assigns to every PCF type σ some mathematical structure $[\![\sigma]\!]$ and to every PCF term t of type σ an element $[\![t]\!]$ of $[\![\sigma]\!]$. In Scott's model [Sco93], we interpret the PCF types as pointed dcpos. Specifically, in our constructive and predicative setting, we interpret the base type ι as the lifting of the type of natural numbers and function types using exponentials. The least element serves as an interpretation of a nonterminating computation, as is made precise by computational adequacy discussed below. As mentioned in the introduction to this chapter, the fact that Scott continuous maps have least fixed points (Theorem 3.5.9) will be fundamental in giving a sound meaning to PCF's fixed point combinator.

Soundness and computational adequacy are important properties that any model of PCF should have. Soundness states that if a PCF term s reduces to t (according to the operational semantics), then their interpretations are equal in the model (symbolically, $[\![s]\!] = [\![t]\!]$). Computational adequacy is completeness at the base type ι. It says that for every term t of type ι and every natural number n, if $[\![t]\!] = [\![\underline{n}]\!]$, then t reduces to $\underline{n}$. The Scott model of PCF was originally proved to be sound and computationally adequate by Plotkin [Plo77] and we prove these results in our foundational type-theoretic setup too.

Since the base type is interpreted using the lifting, every PCF term of the base type is interpreted as a partial element of $\mathbb{N}$, and hence gives rise to a proposition (the domain of the partial element). Motivated by constructive issues involving countable choice (see Section 5.2.4), we use computational adequacy in a syntactic approach to establishing that all such propositions are semidecidable.

5.2.1 PCF and its operational semantics

We precisely define the types and terms of PCF as well as the small-step operational semantics. Instead of the formulation by Plotkin [Plo77], which features variables and λ-abstraction, we revert to the original, combinatory, formulation of the terms of LCF by Scott [Sco93] in order to simplify the technical development. We note that it is possible to represent every closed λ-term in terms of combinators by a well-known technique [HP08, Section 2C].

Definition 5.2.1 (PCF types, ι, $\sigma \Rightarrow \tau$)**.** The *PCF types* are inductively defined as

(i) ι is a type, the *base type*, and

(ii) for every two types σ and τ, there is a *function type* $\sigma \Rightarrow \tau$.

As usual, $\Rightarrow$ will be right associative, so we write $\sigma \Rightarrow \tau \Rightarrow \rho$ for $\sigma \Rightarrow (\tau \Rightarrow \rho)$.

Definition 5.2.2 (PCF terms, $\mathtt{zero}$, $\mathtt{succ}$, $\mathtt{pred}$, $\mathtt{ifz}$, $\mathtt{k}_{\sigma,\tau}$, $\mathtt{s}_{\sigma,\tau,\rho}$, $\mathtt{fix}_\sigma$)**.** The *PCF terms of PCF type* σ are inductively generated by

$$\frac{}{\mathtt{zero}\text{ of type }\iota} \qquad \frac{}{\mathtt{succ}\text{ of type }\iota \Rightarrow \iota}$$

$$\frac{}{\mathtt{pred}\text{ of type }\iota \Rightarrow \iota} \qquad \frac{}{\mathtt{ifz}\text{ of type }\iota \Rightarrow \iota \Rightarrow \iota \Rightarrow \iota}$$

$$\frac{}{\mathtt{k}_{\sigma,\tau}\text{ of type }\sigma \Rightarrow \tau \Rightarrow \sigma} \qquad \frac{}{\mathtt{s}_{\sigma,\tau,\rho}\text{ of type }(\sigma \Rightarrow \tau \Rightarrow \rho) \Rightarrow (\sigma \Rightarrow \tau) \Rightarrow \sigma \Rightarrow \rho}$$

$$\frac{}{\mathtt{fix}_\sigma\text{ of type }(\sigma \Rightarrow \sigma) \Rightarrow \sigma} \qquad \frac{s\text{ of type }\sigma \Rightarrow \tau \qquad t\text{ of type }\sigma}{(st)\text{ of type }\tau}$$

We will often drop the parentheses in the final clause, as well as the PCF type subscripts in $\mathtt{k}_{\sigma,\tau}$, $\mathtt{s}_{\sigma,\tau,\rho}$ and $\mathtt{fix}_\sigma$. Finally, we employ the convention that the parentheses associate to the left, i.e. we write rst for $(rs)t$.

Definition 5.2.3 (PCF numerals, $\underline{n}$)**.** For any $n : \mathbb{N}$, let us write $\underline{n}$ for the n^{th} PCF numeral, defined inductively by $\underline{0} :\equiv \mathtt{zero}$ and $\underline{n+1} :\equiv \mathtt{succ}\ \underline{n}$.

To define the small-step operational semantics of PCF, we first define an auxiliary type family by induction and then we propositional truncate its values to obtain the small-step relation.

Definition 5.2.4 (Small-step (pre-)relation, $\tilde{\triangleright}$, $\triangleright$)**.** Define the *small-step pre-relation* $\tilde{\triangleright}$ of type

$$\Pi_{\sigma:\text{PCF types}}(\text{PCF terms of type }\sigma \to \text{PCF terms of type }\sigma \to \mathcal{U}_0)$$

as the inductive family generated by

$$\frac{}{\mathtt{pred}\,\underline{0}\mathrel{\tilde{\triangleright}}\underline{0}} \qquad \frac{}{\mathtt{pred}\,\underline{n+1}\mathrel{\tilde{\triangleright}}\underline{n}} \qquad \frac{}{\mathtt{ifz}\,s\,t\,\underline{0}\mathrel{\tilde{\triangleright}} s} \qquad \frac{}{\mathtt{ifz}\,s\,t\,\underline{n+1}\mathrel{\tilde{\triangleright}} t}$$

$$\frac{}{\mathtt{k}\, s\, t \mathrel{\tilde{\triangleright}} s} \qquad \frac{}{\mathtt{s} f\, g\, t \mathrel{\tilde{\triangleright}} f\, t (g\, t)} \qquad \frac{}{\mathtt{fix}\, f \mathrel{\tilde{\triangleright}} f(\mathtt{fix}\, f)} \qquad \frac{f \mathrel{\tilde{\triangleright}} g}{f\, t \mathrel{\tilde{\triangleright}} g\, t}$$

$$\frac{s \mathrel{\tilde{\triangleright}} t}{\mathtt{succ}\, s \mathrel{\tilde{\triangleright}} \mathtt{succ}\, t} \qquad \frac{s \mathrel{\tilde{\triangleright}} t}{\mathtt{pred}\, s \mathrel{\tilde{\triangleright}} \mathtt{pred}\, t} \qquad \frac{r \mathrel{\tilde{\triangleright}} r'}{\mathtt{ifz}\, s\, t\, r \mathrel{\tilde{\triangleright}} \mathtt{ifz}\, s\, t\, r'}$$

We have been unable to prove that $s \mathrel{\tilde{\triangleright}} t$ is a proposition for all PCF terms s and t of the same type. The difficulty is that one cannot perform induction on *both* s and t, because the reduction relation is not defined by induction on terms. However, conceptually, $s \mathrel{\tilde{\triangleright}} t$ should be a proposition, as (by inspection of the definition), there is at most one way by which we obtained $s \mathrel{\tilde{\triangleright}} t$. Moreover, for technical reasons that will become apparent later, we need $\tilde{\triangleright}$ to be proposition-valued.
We solve the problem by defining the *small-step relation* $\triangleright$ as the propositional truncation of $\tilde{\triangleright}$, i.e. $s \triangleright t :\equiv \|s \mathrel{\tilde{\triangleright}} t\|$.

Remark 5.2.5. Benedikt Ahrens pointed out that in an impredicative framework, one could use propositional resizing and an impredicative encoding, i.e. by defining $\triangleright$ as a Π-type of all suitable proposition-valued relations. This is similar to the situation in set theory, where one would define $\triangleright$ as an intersection. Specifically, say that a relation

$$R : \Pi_{\sigma:\text{PCF types}}(\text{PCF terms of type } \sigma \to \text{PCF terms of type } \sigma \to \Omega_{\mathcal{U}_0})$$

is *suitable* if it closed under all the clauses of Definition 5.2.4, that is, we would want to have elements of

$$R(\iota, \mathtt{pred}\, \underline{0}, \underline{0}), R(\iota, \mathtt{pred}\, \underline{n+1}, \underline{n}), R(\iota, \mathtt{ifz}\ s\, t\, \underline{0}, s), \text{etc.}$$

We could then define $s \triangleright_{\text{impred}} t :\equiv \Pi_{R \text{ suitable}} R(\sigma, s, t)$. But notice the increase in universes:

$$\triangleright_{\text{impred}} : \Pi_{\sigma:\text{PCF types}}(\text{PCF terms of type } \sigma \to \text{PCF terms of type } \sigma \to \Omega_{\mathcal{U}_1}).$$

So because of this increase, $\triangleright_{\text{impred}}$ itself is not one of the suitable relations. Moreover, the definition $\triangleright_{\text{impred}}$ only quantifies over $\mathcal{U}_0$-valued relations, and so has no bearing on relations valued in other universes. Therefore $\triangleright_{\text{impred}}$ does not satisfy the appropriate universal property of being the least relation closed under the clauses in Definition 5.2.4. Notice that this is exactly the same phenomenon as discussed in Section 2.11.1 for the Voevodsky propositional truncation. With propositional resizing we could resize the relations to all have values in $\mathcal{U}_0$ and obtain the appropriate universal property. The advantage of using the propositional truncation above is that it does satisfy the right universal property even without propositional resizing.

Let $R : X \to X \to \Omega$ be a relation on a type X. We might try to define the reflexive transitive closure R_* of R as an inductive family, generated by three constructors:

$$\begin{aligned} \text{extend} &: \Pi_{x,y:X}(x\ R\ y \to x\ R_*\ y); \\ \text{refl} &: \Pi_{x:X}(x\ R_*\ x); \\ \text{trans} &: \Pi_{x,y,z:X}(x\ R_*\ y \to y\ R_*\ z \to x\ R_*\ z). \end{aligned}$$

But R_* is not necessarily proposition-valued, even though R is. This is because we might add a pair (x, y) to R_* in more than one way, for example, once by an instance of extend and once by an instance of trans. Thus, we are led to the following definition.

Definition 5.2.6. Let $R : X \to X \to \Omega$ be a relation on a type X. We define the *reflexive transitive closure* R^* of R by $x\ R^*\ y :\equiv \|x\ R_*\ y\|$, where R_* is as above.

It is not hard to show that R^* is the least reflexive and transitive proposition-valued relation that extends R, so R^* satisfies the appropriate universal property.

Some properties of $\triangleright$ reflect onto $\triangleright^*$ as the following lemma shows.

Lemma 5.2.7. *Let r', r, s and t be PCF terms of type ι. If $r' \triangleright^* r$, then*

(i) $\mathtt{succ}\, r' \triangleright^* \mathtt{succ}\, r$,

(ii) $\mathtt{pred}\, r' \triangleright^* \mathtt{pred}\, r$, *and*

(iii) $\mathtt{ifz}\, s\, t\, r' \triangleright^* \mathtt{ifz}\, s\, t\, r$.

Moreover, if f and g are PCF terms of type $\sigma \Rightarrow \tau$ and $f \triangleright^* g$, then $f t \triangleright^* g t$ for any PCF term t of type σ.

Proof. We only prove (i) the rest is similar. Suppose $r' \triangleright^* r'$. Since $\mathtt{succ}\, r' \triangleright^* \mathtt{succ}\, r$ is a proposition, we may assume that we actually have an element p of type $r' \triangleright_* r'$. Now we can perform induction on p. The cases were p is formed using refl or trans are easy. If p is formed by extend, then we get an element of type $r \triangleright r' \equiv r \tilde{\triangleright} r'$. Again, as we are proving a proposition, we may suppose the existence of an element of type $r \tilde{\triangleright} r'$. By Definition 5.2.4, we then get $\mathtt{succ}\, r \tilde{\triangleright} \mathtt{succ}\, r'$. This in turn yields, $\mathtt{succ}\, r' \triangleright \mathtt{succ}\, r$ and finally we use extend to get the desired $\mathtt{succ}\, r' \triangleright^* \mathtt{succ}\, r$. □

5.2.2 The Scott model of PCF

We proceed by defining our constructive version of the Scott model of PCF using pointed (exponentials of) $\mathcal{U}_0$-dcpos. Recall from Proposition 3.4.1 that interpreting the base type by naively adding a least element to the type of natural numbers is constructively inadequate, which is why we use the lifting (as defined in Section 3.4) of the type of natural numbers.

Definition 5.2.8 (Interpretation of PCF types, $[\![\sigma]\!]$). We inductively define a map

$$[\![-]\!] : \text{PCF types} \to \mathcal{U}_0\text{-DCPO}_{\mathcal{U}_1,\mathcal{U}_1}$$

interpreting a PCF type as a pointed $\mathcal{U}_0$-dcpo as follows:

(i) $[\![\iota]\!] :\equiv \mathcal{L}(\mathbb{N})$;

(ii) $[\![\sigma \Rightarrow \tau]\!] :\equiv [\![\tau]\!]^{[\![\sigma]\!]}$.

We recall Remark 3.5.7 on universe parameters and we remark that is fortunate that the interpretation function $[\![-]\!]$ takes values in $\mathcal{U}_0$-DCPO$_{\mathcal{U}_1,\mathcal{U}_1}$ and does not require ever-increasing universe parameters.

Next, we interpret PCF terms as elements of these pointed dcpos, for which we will need that $\mathcal{L}$ is a monad (with unit η) and (in particular) a functor (recall Theorem 3.4.9 and Definition 3.4.12).

Definition 5.2.9 (Interpretation of PCF terms, $[\![t]\!]$)**.** Define for each PCF term t of PCF type σ a term $[\![t]\!]$ of type $[\![\sigma]\!]$, by the following inductive clauses:

(i) $[\![\texttt{zero}]\!] :\equiv \eta(0)$;

(ii) $[\![\texttt{succ}]\!] :\equiv \mathcal{L}(s)$, where $s : \mathbb{N} \to \mathbb{N}$ is the successor function;

(iii) $[\![\texttt{pred}]\!] :\equiv \mathcal{L}(p)$, where $p : \mathbb{N} \to \mathbb{N}$ is the predecessor function with the convention that 0 is mapped to 0;

(iv) $[\![\texttt{ifz}]\!] : [\![\iota \Rightarrow \iota \Rightarrow \iota \Rightarrow \iota]\!]$ is defined using the Kleisli extension (Theorem 3.4.9) as: $\lambda x, y.\left(\chi_{x,y}\right)^{\#}$, where

$$\chi_{x,y}(n) :\equiv \begin{cases} x & \text{if } n = 0; \\ y & \text{else;} \end{cases}$$

(v) $[\![\texttt{k}]\!] :\equiv \lambda x, y.x$;

(vi) $[\![\texttt{s}]\!] :\equiv \lambda f, g, x.(f(x))(g(x))$;

(vii) $[\![\texttt{fix}]\!] :\equiv \mu$, where μ is the least fixed point operator from Theorem 3.5.9;

(viii) $[\![st]\!] :\equiv [\![s]\!]([\![t]\!])$ for s of type $\sigma \Rightarrow \tau$ and t of type σ.

Remark 5.2.10. Of course, there are some things to be proved here. Namely, $[\![\texttt{succ}]\!]$, $[\![\texttt{pred}]\!]$, ..., $[\![\texttt{fix}]\!]$, $[\![s\,t]\!]$ all need to be Scott continuous. In the case of $[\![\texttt{succ}]\!]$ and $[\![\texttt{pred}]\!]$, we simply appeal to Proposition 3.4.15 and Definition 3.4.12. For $[\![\texttt{fix}]\!]$, this is guaranteed by Theorem 3.5.9. The continuity of $[\![\texttt{k}]\!]$, $[\![\texttt{s}]\!]$ and $[\![\texttt{ifz}]\!]$ can be verified directly; the details are omitted here. Finally, the interpretation of application is continuous by Proposition 3.5.8.

As a first result about our denotational semantics, we show that the PCF numerals have a canonical interpretation in the denotational semantics. This basic result is fundamental and finds application in the proof of soundness.

Proposition 5.2.11. *For every natural number n, we have $[\![\underline{n}]\!] = \eta(n)$.*

Proof. We proceed by induction on n. The $n \equiv 0$ case is by definition of $[\![\underline{0}]\!]$. Suppose $[\![\underline{m}]\!] = \eta(m)$ for a natural number m. Then,

$$\begin{aligned} [\![\underline{m+1}]\!] &\equiv [\![\texttt{succ}]\!]([\![\underline{m}]\!]) \\ &= \mathcal{L}(s)(\eta(m)) && \text{(by induction hypothesis)} \\ &= \eta(m+1) && \text{(by definition of the lift functor),} \end{aligned}$$

as desired. □

5.2.3 Soundness and computational adequacy

Having defined the Scott model of PCF we show that it respects the operational semantics by proving soundness and computational adequacy.

Theorem 5.2.12 (Soundness). *If $s \rhd^* t$, then $[\![s]\!] = [\![t]\!]$ for every two PCF terms s and t (necessarily of the same type).*

Proof. Since the carriers of dcpos are sets, the type $[\![s]\!] = [\![t]\!]$ is a proposition. Therefore, we can use induction on the derivation of $s \rhd^* t$. We use the Kleisli monad laws in proving some of the cases. For example, one step is to prove that

$$[\![\mathtt{ifz}\, s\, t\, \underline{n+1}]\!] = [\![t]\!].$$

This may be proved by the following chain of equalities:

$$\begin{aligned}
[\![\mathtt{ifz}\, s\, t\, \underline{n+1}]\!] &\equiv [\![\mathtt{ifz}\, s\, t]\!]([\![\underline{n+1}]\!]) \\
&= [\![\mathtt{ifz}\, s\, t]\!](\eta(n+1)) && \text{(by Proposition 5.2.11)} \\
&\equiv \left(\chi_{[\![s]\!],[\![t]\!]}\right)^{\#}(\eta(n+1)) && \text{(by definition of } [\![\mathtt{ifz}]\!]) \\
&= \chi_{[\![s]\!],[\![t]\!]}(n+1) && \text{(by Theorem 3.4.9)} \\
&= [\![t]\!].
\end{aligned}$$

The other cases are proved similarly. □

Ideally, we would like a converse to soundness. However, this is not possible, as for example, $[\![\mathtt{k}\,\mathtt{zero}]\!] = [\![\mathtt{k}(\mathtt{succ}(\mathtt{pred}\,\mathtt{zero}))]\!]$, but neither $\mathtt{k}\,\mathtt{zero} \rhd^* \mathtt{k}(\mathtt{succ}(\mathtt{pred}\,\mathtt{zero}))$ nor $\mathtt{k}(\mathtt{succ}(\mathtt{pred}\,\mathtt{zero})) \rhd^* \mathtt{k}\,\mathtt{zero}$ holds. We do, however, have the following.

Theorem 5.2.13 (Computational adequacy). *For a PCF term t of the base type, if the partial element $[\![t]\!]$ is defined, then t reduces to the numeral given by the value of $[\![t]\!]$.*

Equivalently, for every $n : \mathbb{N}$, it holds that $[\![t]\!] = [\![\underline{n}]\!]$ implies $t \rhd^* \underline{n}$. Another useful rephrasing is: for every PCF term t of the base type, we have $t \rhd^* \underline{\mathrm{value}([\![t]\!], p)}$ for every $p : \text{is-defined}([\![t]\!])$.

We do not prove computational adequacy directly, as, unlike soundness, it does not allow for a straightforward proof by induction on terms. Instead, we use the standard technique of logical relations [Str06, Chapter 7] which goes back to [Tai67] and obtain the result as a direct corollary of Lemma 5.2.20.

Definition 5.2.14 (Logical relation, R_σ). For every PCF type σ, define a relation

$$R_\sigma \colon \text{PCF terms of type } \sigma \to [\![\sigma]\!] \to \Omega_{\mathcal{U}_0}$$

by induction on σ as

(i) $t\ R_\iota\ d :\equiv \Pi_{p:\text{is-defined}(d)}\left(t \rhd^* \underline{\mathrm{value}(d,p)}\right)$, and

(ii) $s\ R_{\tau\Rightarrow\rho}\ f :\equiv \Pi_{t:\text{PCF terms of type }\tau}\Pi_{d:[\![\tau]\!]}\big(t\ R_\tau\ d \to st\ R_\rho\ f(d)\big)$.

We sometimes omit the type subscript σ in R_σ.

Note that (i) in Definition 5.2.14 is the statement of computational adequacy. By generalising, we can prove properties of R by induction on types.

Lemma 5.2.15. *If $s \triangleright^* t$ and $t \; R_\sigma \; d$, then $s \; R_\sigma \; d$, for all PCF types σ and PCF terms s and t of type σ and elements $d : [\![\sigma]\!]$.*

Proof. By induction on σ, making use of the last part of Lemma 5.2.7. □

Lemma 5.2.16. *For t equal to $\texttt{zero}$, $\texttt{succ}$, $\texttt{pred}$, $\texttt{ifz}$, $\texttt{k}$ or $\texttt{s}$, we have $t \; R \; [\![t]\!]$.*

Proof. By the previous lemma and Lemma 5.2.7. □

Next, we wish to extend the previous lemma to the case where $t \equiv \texttt{fix}_\sigma$ for any PCF type σ. This is slightly more complicated and we need two intermediate lemmas.

Lemma 5.2.17. *For every PCF type σ and term t of type σ it holds that $t \; R_\sigma \perp$.*

Proof. By induction on σ: for the base type, this holds vacuously; for function types, it follows by induction hypothesis and the pointwise ordering. □

Lemma 5.2.18. *The logical relation is closed under directed suprema. That is, for every PCF type σ and every PCF term t of type σ and every directed family $d : I \longrightarrow [\![\sigma]\!]$, if $t \; R_\sigma \; d_i$ for every $i : I$, then $t \; R_\sigma \bigsqcup_{i:I} d_i$.*

Proof. This proof is somewhat different from the classical proof, so we spell out the details. We prove the lemma by induction on σ. The case when σ is a function type is easy, because least upper bounds are calculated pointwise and so it reduces to an application of the induction hypothesis. We concentrate on the case when $\sigma \equiv \iota$ instead. Recall that $\bigsqcup_{i:I} d_i$ is given by $(\exists_{i:I} \, \text{is-defined}(d_i), \phi)$, where ϕ is the factorisation of

$$\left(\Sigma_{i:I} \, \text{is-defined}(d_i)\right) \xrightarrow{(i,p) \mapsto \text{value}(d_i,p)} \mathcal{L}(\mathbb{N})$$

through $\exists_{i:I} \, \text{is-defined}(d_i)$. We are tasked with proving that $t \triangleright^* \underline{\phi(p)}$ whenever p witnesses that $\bigsqcup_{i:I} d_i$ is defined. Since we are trying to prove a proposition (as $\triangleright^*$ is proposition-valued), we may assume that we have $(j,p) : \Sigma_{i:I} \, \text{is-defined}(d_i)$. By definition of ϕ we have: $\phi(|(j,p)|) = \text{value}(d_j,p)$ and by assumption we know that $t \triangleright^* \underline{\text{value}(d_j,p)}$, so we are done. □

Lemma 5.2.19. *For every PCF type σ, we have $\texttt{fix}_\sigma \; R_{(\sigma \Rightarrow \sigma) \Rightarrow \sigma} \; [\![\texttt{fix}_\sigma]\!]$.*

Proof. Suppose that $t \; R_{\sigma \Rightarrow \sigma} \; f$; we are to prove that $\texttt{fix}\, t \; R_\sigma \; \mu(f)$. By the previous lemma, it suffices to prove that $\texttt{fix}\, t \; R_\sigma \; f^n(\perp)$ for every natural number n, which we do by induction on n. The base case is an application of Lemma 5.2.17. Now suppose that $\texttt{fix}\, t \; R_\sigma \; f^m(\perp)$. Then, using $t \; R_{\sigma \Rightarrow \sigma} \; f$, we find $t(\texttt{fix}\, t) \; R_\sigma \; f(f^m(\perp))$. Hence, by Lemma 5.2.15, we obtain the $\texttt{fix}\, t \; R_\sigma \; f^{m+1}(\perp)$, completing our inductive proof. □

Lemma 5.2.20 (Fundamental Theorem)**.** *We have $t\ R\ [\![t]\!]$ for every PCF term t.*

Proof. The proof is by induction on t. The base cases are taken care of by Lemma 5.2.16 and the previous lemma. For the inductive step, suppose t is a PCF term of type $\sigma \Rightarrow \tau$. By induction hypothesis, $ts\ R_\tau\ [\![ts]\!]$ for every PCF term s of type σ, but $[\![ts]\!] \equiv [\![t]\!][\![s]\!]$, so we are done. □

Computational adequacy is now a direct corollary of Lemma 5.2.20.

Proof of computational adequacy. Take σ to be the base type ι in Lemma 5.2.20. □

Using computational adequacy to compute. An interesting use of computational adequacy is that it allows one to argue semantically to obtain results about termination (i.e. reduction to a numeral) in PCF. Classically, every PCF program of type ι either terminates or it does not. From a constructive point of view, we wait for a program to terminate, with no a priori knowledge of termination. The waiting could be indefinite. Less naively, we could limit the number of computation steps to avoid indefinite waiting, with an obvious shortcoming: how many steps are enough? Instead, one could use computational adequacy to compute as we describe now.

For a PCF type σ, a *functional of type σ* is an element of $[\![\sigma]\!]$. By induction on PCF types, we define when a functional is said to be *total*:

(i) a functional i of type ι is total if $i = [\![\underline{n}]\!]$ for some natural number n;
(ii) a functional f of type $\sigma \Rightarrow \tau$ is total if it maps total functionals to total functionals, viz. $f(d)$ is a total functional of type τ for every total functional d of type σ.

Now, let s be a PCF term of type $\sigma_1 \Rightarrow \sigma_2 \Rightarrow \cdots \Rightarrow \sigma_n \Rightarrow \iota$. If we can prove that $[\![s]\!]$ is total, then computational adequacy lets us conclude that for all total inputs $[\![t_1]\!] : [\![\sigma_1]\!], \ldots, [\![t_n]\!] : [\![\sigma_n]\!]$, the term $s(t_1, \ldots, t_n)$ reduces to the numeral representing $[\![s]\!]([\![t_1]\!], \ldots, [\![t_n]\!])$. Thus, instead of e.g. giving a number of steps as a timeout for the computation, we supply a proof of totality to computational adequacy and we are guaranteed to obtain a result.

Of course, this approach still requires us to prove that $[\![s]\!]$ is total, which may be challenging. But note that we can use domain-theoretic arguments to prove this about the denotation $[\![s]\!]$, whereas in a direct proof of termination we would only have the operational semantics available for our argument.

5.2.4 Semidecidability and PCF terms of the base type

Given a PCF term t of the base type, we intuitively expect it to be semidecidable whether t will compute to a numeral, as we can reduce t one step at a time following the operational semantics of PCF and stop when we have obtained a numeral.

A rather slick way of proving this would be to argue that we could have restricted to semidecidable propositions in the lifting of the natural numbers, i.e. we modify the Scott model of PCF and set $[\![\iota]\!] :\equiv \mathcal{L}_{\mathrm{sd}}(\mathbb{N})$ with

$$\mathcal{L}_{\mathrm{sd}}(X) :\equiv \Sigma_{P:\Omega_{\mathcal{U}_0}}(P \text{ is semidecidable}) \times (P \to X).$$

This restricted lifting of a set does not necessarily yield a dcpo, but observe that only used suprema of ω-chains in the Scott model of PCF. Thus it would suffice for $\mathcal{L}_{\mathrm{sd}}(X)$ to be a ω-cpo. Escardó and Knapp [EK17, Corollary 5] observed that this is indeed the case *if* countable choice is assumed. And in this case it coincides with the other constructions [CUV19; ADK17] of ω-cpos, as discussed in the Notes of Chapter 3. Countable choice is used to obtain witnessing sequences α_n for a given $\mathbb{N}$-indexed sequence P_n of semidecidable propositions. Indeed, at least some weak form of countable choice is indeed necessary for $\mathcal{L}_{\mathrm{sd}}(X)$ to be closed under countable joins as shown by [dJon22c]. But since countable choice is not provable in constructive univalent foundations (cf. the Notes of Chapter 3), we resort to a different, syntactic approach. That is, without using countable choice, we use soundness and computational adequacy to prove that is-defined($[\![t]\!]$) is semidecidable for every PCF term of the base type. In other words, although we cannot define the Scott model using $\mathcal{L}_{\mathrm{sd}}$, the map

$$[\![-]\!] : \text{PCF terms of the base type} \to \mathcal{L}(\mathbb{N})$$

factors through $\mathcal{L}_{\mathrm{sd}}(\mathbb{N})$, even in the absence of countable choice.

Specifically, we prove semidecidability of the proposition is-defined($[\![t]\!]$) by appealing to Lemma 2.7.14 and showing that is-defined($[\![t]\!]$) is logically equivalent to $\exists_{n:\mathbb{N}}\exists_{k:\mathbb{N}}\, t \triangleright^k \underline{n}$, where $t \triangleright^k \underline{n}$ says that t reduces to $\underline{n}$ in at most k steps. We prove this notion to be decidable by using that the terms of PCF have decidable equality which is an application of the theory on indexed W-types developed in Section 2.12. It turns out to be helpful to study the k-step reflexive transitive closure of an arbitrary relation more generally and isolate criteria for it to be decidable.

Decidability of the k-step reflexive transitive closure of a relation

Fix an arbitrary type X and a proposition-valued binary relation R on X. We define the k-step reflexive transitive closure of R. As in Definition 5.2.6, we want this relation to be proposition-valued again, so we proceed with an auxiliary definition that we truncate.

Definition 5.2.21 (k-step reflexive transitive closure, R_k, R^k). Define $x\ R_k\ y$ by induction on the natural number k as

(i) $x\ R_0\ y :\equiv x = y$, and
(ii) $x\ R_{k+1}\ z :\equiv \Sigma_{y:X}(x\ R\ y) \times (y\ R_k\ z)$.

The *k-step reflexive transitive closure* R^k of R is given by propositionally truncating: $x\ R^k\ y :\equiv \|x\ R_k\ y\|$.

The following proposition relates the reflexive transitive closure of R and its k-step reflexive transitive closure.

Proposition 5.2.22. *For every x and y we have $x\ R_*\ y$ if and only if $\Sigma_{k:\mathbb{N}}(x\ R_k\ y)$, where R_* is the untruncated reflexive transitive closure from just before Definition 5.2.6. Hence, for every x and y it holds that $x\ R^*\ y$ if and only if $\exists_{k:\mathbb{N}}(x\ R^k\ y)$.*

Proof. We define the auxiliary binary relation R' on X inductively: for every $x : X$ we have $x\ R'\ x$; and if $x\ R\ y$ and $y\ R'\ z$, then $x\ R'\ z$. Then R' is reflexive, transitive and it extends R. It follows that R' and R_* are equivalent, so it remains to show that

$x\ R'\ y$ and $\Sigma_{k:\mathbb{N}}(x\ R_k\ y)$ are logically equivalent. In one direction, induction on k yields a proof that $x\ R_k\ y$ implies $x\ R'\ y$ for every natural number k, and hence that $\Sigma_{k:\mathbb{N}}(x\ R_k\ y)$ implies $x\ R'\ y$. The other direction is obtained by induction on the constructors of R'.
The final claim follows from the functoriality of the truncation and the general fact that $\|\Sigma_{x:X}A(x)\|$ and $\|\Sigma_{x:X}\|A(x)\|\|$ are equivalent [Uni13, Theorem 7.3.9]. □

Definition 5.2.23 (Singe-valuedness and decidability of a relation)**.** The relation R is said to be

(i) *single-valued* if for every x, y and z with $x\ R\ y$ and $x\ R\ z$, we have $y = z$, and
(ii) *decidable* if the type $x\ R\ y$ is decidable for every x and y in X.

Proposition 5.2.24. *The k-step reflexive transitive closure R^k is decidable for every natural number k if*

(i) the type X has decidable equality,
(ii) the relation R is single-valued, and
(iii) the type $\Sigma_{y:X}(x\ R\ y)$ is decidable for every $x : X$.

Proof. Since the propositional truncation of a type is decidable as soon the type itself is, it suffices to prove that the relation R_k is decidable, which we do by induction on k. For $k = 0$, this means decidability of $x = y$ for every x and y which we have by assumption (i). Now suppose that R^k is decidable and let x and z be arbitrary elements of X. We prove that $x\ R_{k+1}\ z$ is decidable. By definition, this means proving that

$$\Sigma_{y:X}(x\ R\ y) \times (y\ R_k\ z) \tag{†}$$

is decidable. Using (iii) we can decide whether $\Sigma_{y:X}(x\ R\ y)$ has an element or not. If it does not, then obviously (†) has no elements either. So assume that we have $y : X$ with $x\ R\ y$. By induction hypothesis, the type $y\ R_k\ z$ is decidable. If it has an element, then so does (†). If it does not, then we claim that (†) must be empty too. For if it isn't, then we get $y' : X$ with $x\ R\ y'$ and $y'\ R_k\ z$. But the relation R is assumed to be single-valued, so $y' = y$ and hence $y\ R_k\ z$, contradicting our assumption. □

Semidecidability at the base type

After completing the generalities above, we are now ready to complete the proof of the strategy outlined at the start of this section. The application of Proposition 5.2.24 to the small-step reduction relation of PCF requires us to prove that the syntax of PCF has decidable equality, which follows from the results on indexed W-types featured in Section 2.12.

Theorem 5.2.25. *For every PCF term t of the base type, the proposition* is-defined($[\![t]\!]$) *is semidecidable as witnessed by the logical equivalence*

$$\text{is-defined}([\![t]\!]) \iff \exists_{n:\mathbb{N}}\exists_{k:\mathbb{N}}\, t \triangleright^k \underline{n}$$

and the decidability of $t \triangleright^k \underline{n}$.

Proof. First of all, observe that $[\![t]\!]$ is defined if and only if there exists $n : \mathbb{N}$ such that $t \triangleright^* \underline{n}$. Indeed, if we have p : is-defined($[\![t]\!]$), then $t \triangleright^* \underline{\mathrm{value}([\![t]\!], p)}$ by computational adequacy. Conversely, if there exists a natural number n such that $t \triangleright^* \underline{n}$, then $[\![t]\!] = \eta(n)$ by soundness and Proposition 5.2.11, so that $[\![t]\!]$ must be defined. Furthermore, by Proposition 5.2.22, we have that $t \triangleright^* \underline{n}$ is logically equivalent to $\exists_{k:\mathbb{N}}\, t \triangleright^k \underline{n}$, so this proves the logical equivalence is-defined($[\![t]\!]$) $\iff \exists_{n:\mathbb{N}}\exists_{k:\mathbb{N}}\, t \triangleright^k \underline{n}$. Hence, it only remains to prove that $t \triangleright^k \underline{n}$ is decidable, for which we use Proposition 5.2.24. Accordingly, we need to check its three conditions. First of all, the type of PCF terms should have decidable equality, which is guaranteed by a modest extension of Proposition 2.12.14. The other two conditions can be proved by inspection of the operational semantics of PCF using decidability of equality of PCF terms. □

5.3 Notes

The treatment of Scott's D_∞ model of the untyped λ-calculus in Section 5.1 is an expanded account of Section 5.2 of our paper [dJE21a], while Section 5.2 is a slight revision of the exposition in our publication [dJon21b].

The proof that D_∞ is isomorphic to $D_\infty^{D_\infty}$ largely follows that of [Sco72, Theorem 4.4], although we instantiate the general framework involving directed bilimits set out in Section 3.6, rather than working with sequential bilimits directly.

The Scott model was proved sound and computationally adequate by Plotkin [Plo77], and the techniques of Scott and Plotkin have been extended to many other programming languages [Plo83]. Our proof follows the modern presentation given by Streicher [Str06], although, instead of formulating PCF with variables and λ-abstraction, we revert to the original, combinatory, formulation of the terms of LCF by Scott [Sco93] in order to simplify the technical development.

The formulation of computational adequacy in terms of is-defined($[\![t]\!]$), and the suggestion that it could be leveraged to prove semidecidability of these propositions are due to Martín Escardó.

Chapter 6

Predicativity in order theory

In the preceding chapters we gave a type-theoretic account of constructive and predicative domain theory including many familiar constructions and notions, such as Scott's D_∞ model of the untyped λ-calculus and the theory of continuous dcpos. In this chapter we complement this by exploring what cannot be done predicatively.

6.1 Introduction

The work in this chapter is in the spirit of constructive reverse mathematics [Ish06] and amounts to predicative reverse mathematics: we show certain statements to crucially rely on resizing axioms in the sense that they are equivalent to them. Such arguments are important in constructive mathematics. For example, the constructive failure of trichotomy on the real numbers is shown [BR87] by reducing it to a nonconstructive instance of excluded middle. As another example, note that in Proposition 3.4.1 we used a reduction to the limited principle of omniscience (LPO) to show that $\mathbb{N}_\perp$ cannot be a dcpo constructively.

Our first main result is that nontrivial (directed or bounded) complete posets are necessarily large. All examples of dcpos that we have seen have large carriers, in the sense that all examples of $\mathcal{V}$-dcpos have carriers that live in $\mathcal{V}^+$ or some higher universe. We show here that this is no coincidence, but rather a necessity, in the sense that if such a nontrivial poset is small, then weak propositional resizing holds. It is possible to derive full propositional resizing if we strengthen nontriviality to positivity in the sense of [Joh84]. The distinction between nontriviality and positivity is analogous to the distinction between nonemptiness and inhabitedness. We prove our results for a general class of posets, which includes directed complete posets, bounded complete posets and sup-lattices, using a technical notion of a $\delta_\mathcal{V}$-complete poset. We also show that nontrivial locally small $\delta_\mathcal{V}$-complete posets necessarily lack decidable equality. Specifically, we can derive weak excluded middle from assuming the existence of a nontrivial locally small $\delta_\mathcal{V}$-complete poset with decidable equality. Moreover, if we assume positivity instead of nontriviality, then we can derive full excluded middle.

The fact that nontrivial dcpos are necessarily large has the important consequence that *Tarski's Theorem* (and similar results, such as Pataraia's Lemma) cannot be applied in nontrivial instances, even though it has a predicative proof. Furthermore, we explain that generalisations of Tarski's Theorem that allow for large structures are provably false. Specifically, we show that the ordinal of ordinals in a univalent universe does not have a maximal element, but does have small suprema in the presence of small set quotients or set replacement, illustrating the abstract theory of Section 2.11.

Finally, we clarify, in our predicative setting, the relation between the traditional definition of sup-lattice that requires suprema for all subsets and our definition that asks for suprema of all small families, further explaining our choice to work with families in our development of domain theory.

6.2 Large posets without decidable equality

A well-known result of Freyd in classical mathematics says that every complete small category is a preorder [Fre64, Exercise D of Chapter 3]. In other words, complete categories are necessarily large and only complete preorders can be small, at least impredicatively. Predicatively, by contrast, we show that many weakly complete posets (including directed complete posets, bounded complete posets and sup-lattices) are necessarily large. We capture these structures by a technical notion of a $\delta_{\mathcal{V}}$-complete poset in Section 6.2.1. In Section 6.2.2 we define when such structures are nontrivial and introduce the constructively stronger notion of positivity. Section 6.2.3 and Section 6.2.4 contain the two fundamental technical lemmas and the main theorems, respectively. Finally, we consider alternative formulations of being nontrivial and positive that ensure that these notions are properties rather than data and shows how the main theorems remain valid, assuming univalence.

6.2.1 $\delta_{\mathcal{V}}$-complete posets

We start by introducing a class of weakly complete posets that we call $\delta_{\mathcal{V}}$-complete posets. The notion of a $\delta_{\mathcal{V}}$-complete poset is a technical and auxiliary notion sufficient to make our main theorems go through. The important point is that many familiar structures (dcpos, bounded complete posets, sup-lattices) are $\delta_{\mathcal{V}}$-complete posets (see Example 6.2.3).

Definition 6.2.1 ($\delta_{\mathcal{V}}$-complete poset, $\delta_{x,y,P}$, $\bigvee \delta_{x,y,P}$)**.** A poset X is $\delta_{\mathcal{V}}$*-complete* for a universe $\mathcal{V}$ if for every pair of elements $x, y : X$ with $x \sqsubseteq y$ and every subsingleton P in $\mathcal{V}$, the family

$$\begin{aligned} \delta_{x,y,P} &: 1 + P \to X \\ &\operatorname{inl}(\star) \mapsto x, \text{ and} \\ &\operatorname{inr}(p) \mapsto y \end{aligned}$$

has a supremum $\bigvee \delta_{x,y,P}$ in X.

Remark 6.2.2 (Classically, every poset is $\delta_{\mathcal{V}}$-complete). Consider a pair of elements $x \sqsubseteq y$ of a poset. If $P : \mathcal{V}$ is a decidable proposition, then we can define the supremum of $\delta_{x,y,P}$ by case analysis on whether P holds or not. For if it holds, then the supremum is y, and if it does not, then the supremum is x. Hence, if excluded middle holds in $\mathcal{V}$, then the family $\delta_{x,y,P}$ has a supremum for every $P : \mathcal{V}$. Thus, if excluded middle holds in $\mathcal{V}$, then every poset (with carrier in any universe) is $\delta_{\mathcal{V}}$-complete.

The above remark naturally leads us to ask whether the converse also holds, i.e. if every poset is $\delta_{\mathcal{V}}$-complete, does excluded middle in $\mathcal{V}$ hold? As far as we know, we can only get weak excluded middle in $\mathcal{V}$, as we will later see in Proposition 6.2.6. This proposition also shows that in the absence of excluded middle, the notion of $\delta_{\mathcal{V}}$-completeness isn't trivial. For now, we focus on the fact that, also constructively and predicatively, there are many examples of $\delta_{\mathcal{V}}$-complete posets.

Example 6.2.3 ($\delta_{\mathcal{V}}$-complete posets).

(i) Every $\mathcal{V}$-*sup-lattice* is $\delta_{\mathcal{V}}$-complete. That is, if a poset X has suprema for all families $I \to X$ with I in the universe $\mathcal{V}$, then X is $\delta_{\mathcal{V}}$-complete. Hence, in particular, $\Omega_{\mathcal{V}}$ is $\delta_{\mathcal{V}}$-complete, as is the $\mathcal{V}$-powerset $\mathcal{P}_{\mathcal{V}}(X)$ for a type X in any universe.

(ii) Every $\mathcal{V}$-*bounded complete* poset is $\delta_{\mathcal{V}}$-complete. That is, if X is a poset with suprema for all bounded families $I \to X$ with $I : \mathcal{V}$, then X is $\delta_{\mathcal{V}}$-complete. A family $\alpha : I \to X$ is *bounded* if there exists some $x : X$ with $\alpha(i) \sqsubseteq x$ for every $i : I$. For example, the family $\delta_{x,y,P}$ is bounded by y.

(iii) Every $\mathcal{V}$-dcpo is $\delta_{\mathcal{V}}$-complete, since the family $\delta_{x,y,P}$ is directed.

6.2.2 Nontrivial and positive posets

In Remark 6.2.2 we saw that if we can decide a proposition P, then we can define $\bigvee \delta_{x,y,P}$ by case analysis. What about the converse? That is, if $\delta_{x,y,P}$ has a supremum and we know that it equals x or y, can we then decide P? Of course, if $x = y$, then $\bigvee \delta_{x,y,P} = x = y$, so we don't learn anything about P. But what if we add the assumption that $x \neq y$? It turns out that constructively we can only expect to derive decidability of $\neg P$ in that case. This is due to the fact that $x \neq y$ is a negated proposition, which is rather weak constructively, leading us to later define (see Definition 6.2.8) a constructively stronger notion for elements of $\delta_{\mathcal{V}}$-complete posets.

Definition 6.2.4 (Nontriviality). A poset $(X, \sqsubseteq)$ is *nontrivial* if we have specified $x, y : X$ with $x \sqsubseteq y$ and $x \neq y$.

Lemma 6.2.5. *For a nontrivial poset $(X, \sqsubseteq, x, y)$ and a proposition $P : \mathcal{V}$, we have the following two implications:*

(i) if the supremum of $\delta_{x,y,P}$ exists and $x = \bigvee \delta_{x,y,P}$, then $\neg P$ is the case;

(ii) if the supremum of $\delta_{x,y,P}$ exists and $y = \bigvee \delta_{x,y,P}$, then $\neg\neg P$ is the case.

Proof. Let $P : \mathcal{V}$ be an arbitrary proposition. For (i), suppose that $x = \bigvee \delta_{x,y,P}$ and assume for a contradiction that we have $p : P$. Then $y \equiv \delta_{x,y,P}(\mathrm{inr}(p)) \sqsubseteq \bigvee \delta_{x,y,P} = x$, which is impossible by antisymmetry and our assumptions that $x \sqsubseteq y$ and $x \neq y$. For (ii), suppose that $y = \bigvee \delta_{x,y,P}$ and assume for a contradiction that $\neg P$ holds. Then $x = \bigvee \delta_{x,y,P} = y$, contradicting our assumption that $x \neq y$. □

Proposition 6.2.6. *If the poset* $\mathbf{2}$ *with exactly two elements* $0 \sqsubseteq 1$ *is* $\delta_{\mathcal{V}}$*-complete, then weak excluded middle in* $\mathcal{V}$ *holds.*

Proof. Suppose that $\mathbf{2}$ were $\delta_{\mathcal{V}}$-complete and let $P : \mathcal{V}$ be an arbitrary subsingleton. We must show that $\neg P$ is decidable. Since $\mathbf{2}$ has exactly two elements, the supremum $\bigvee \delta_{0,1,P}$ must be 0 or 1. But then we apply Lemma 6.2.5 to get decidability of $\neg P$. □

Combining Remark 6.2.2 and Proposition 6.2.6 yields that excluded middle implies that every poset is $\delta_{\mathcal{V}}$-complete, which in turns implies weak excluded middle. We do not know whether these implications can be reversed. That the conclusion of the implication in Lemma 6.2.5(ii) cannot be strengthened to say that P must hold is shown by the following observation.

Proposition 6.2.7. *If for every two propositions* Q *and* R *in* $\Omega_{\mathcal{V}}$ *with* $Q \sqsubseteq R$ *and* $Q \neq R$ *we have that the equality* $R = \bigvee \delta_{Q,R,P}$ *in* $\Omega_{\mathcal{V}}$ *implies* P *for every proposition* $P : \mathcal{V}$*, then excluded middle in* $\mathcal{V}$ *follows.*

Proof. Assume the hypothesis in the proposition. We are going to show that $\neg\neg P \to P$ for every proposition $P : \mathcal{V}$, from which excluded middle in $\mathcal{V}$ follows. So let P be a proposition in $\mathcal{V}$ such that its double negation holds. This yields $\mathbf{0} \neq P$, so by assumption the equality $P = \bigvee \delta_{\mathbf{0},P,P}$ implies P. But this equality holds, by construction of suprema in $\Omega_{\mathcal{V}}$. □

Thus, having a pair of elements $x \sqsubseteq y$ with $x \neq y$ is rather weak constructively in that we can only derive $\neg\neg P$ from $y = \bigvee \delta_{x,y,P}$. As promised in the introduction of this section, we now introduce and motivate a constructively stronger notion.

Definition 6.2.8 (Strictly below, $x \sqsubset y$)**.** We say that x is *strictly below* y in a $\delta_{\mathcal{V}}$-complete poset if $x \sqsubseteq y$ and, moreover, for every $z \sqsupseteq y$ and every proposition $P : \mathcal{V}$, the equality $z = \bigvee \delta_{x,z,P}$ implies P.

Note that with excluded middle, $x \sqsubset y$ is equivalent to the conjunction of $x \sqsubseteq y$ and $x \neq y$. But constructively, the former is much stronger, as the following examples and proposition illustrate.

Example 6.2.9 (Strictly below in $\Omega_{\mathcal{V}}$ and $\mathcal{P}_{\mathcal{V}}(X)$)**.**

(i) We illustrate the notion of strictly below in $\Omega_{\mathcal{V}}$. For an arbitrary proposition $P : \mathcal{V}$, we have that $\mathbf{0}_{\mathcal{V}} \neq P$ holds precisely when $\neg\neg P$ does. However, $\mathbf{0}_{\mathcal{V}}$ is strictly below P if and only if P holds. More generally, for any two propositions $Q, P : \mathcal{V}$, we have $(Q \sqsubseteq P) \times (Q \neq P)$ if and only if $\neg Q \times \neg\neg P$ holds. But, $Q \sqsubset P$ holds if and only if $\neg Q \times P$ holds.

(ii) In the powerset $\mathcal{P}_{\mathcal{V}}(X)$ of a type $X : \mathcal{V}$ the situation is slightly more involved, but still illustrative. If we have two subsets $A \sqsubseteq B$ of X, then $A \neq B$ if and only if $\neg(\forall_{x:X}(x \in B \to x \in A))$. However, if $A \sqsubset B$ and $y \in A$ is decidable for every $y : X$, then we get the stronger $\exists_{x:X}(x \in B \times x \notin A)$. For we can take $P : \mathcal{V}$ to be $\exists_{x:X}(x \in B \times x \notin A)$ and observe that $\bigvee \delta_{A,B,P} = B$, because if $x \in B$, either $x \in A$ in which case $x \in \bigvee \delta_{A,B,P}$, or $x \notin A$ in which case P must hold and $x \in B = \bigvee \delta_{A,B,P}$. Conversely, if we have $A \sqsubseteq B$ and an element $x \in B$ with $x \notin A$, then $A \sqsubset B$. For if $C \sqsupseteq B$ is a subset and $P : \mathcal{V}$ a proposition such that $\bigvee \delta_{A,C,P} = C$, then $x \in C = \bigvee \delta_{A,C,P} = A \cup \{y \in C \mid P\}$, so either $x \in A$ or P must hold. But $x \notin A$ by assumption, so P must be true, proving $A \sqsubset B$.

Proposition 6.2.10. *For elements x and y of a $\delta_{\mathcal{V}}$-complete poset, we have that $x \sqsubset y$ implies both $x \sqsubseteq y$ and $x \neq y$. However, if the conjunction of $x \sqsubseteq y$ and $x \neq y$ implies $x \sqsubset y$ for every $x, y : \Omega_{\mathcal{V}}$, then excluded middle in $\mathcal{V}$ holds.*

Proof. Note that $x \sqsubset y$ implies $x \sqsubseteq y$ by definition. Now suppose that $x \sqsubset y$. Then the equality $y = \bigvee \delta_{x,y,\mathbf{0}_{\mathcal{V}}}$ implies that $\mathbf{0}_{\mathcal{V}}$ holds. But if $x = y$, then this equality holds, so $x \neq y$, as desired.
For $P : \Omega_{\mathcal{V}}$ we observed that $\mathbf{0}_{\mathcal{V}} \neq P$ is equivalent to $\neg\neg P$ and that $\mathbf{0}_{\mathcal{V}} \sqsubset P$ is equivalent to P, so if we had $((x \sqsubseteq y) \times (x \neq y)) \to x \sqsubset y$ in general, then we would have $\neg\neg P \to P$ for every proposition P in $\mathcal{V}$, which is equivalent to excluded middle in $\mathcal{V}$. □

Lemma 6.2.11. *The following transitivity properties hold for all elements x, y and z of a $\delta_{\mathcal{V}}$-complete poset:*
(i) if $x \sqsubseteq y \sqsubset z$, then $x \sqsubset z$;
(ii) if $x \sqsubset y \sqsubseteq z$, then $x \sqsubset z$.

Proof. (i) Assume $x \sqsubseteq y \sqsubset z$, let P be an arbitrary proposition in $\mathcal{V}$ and suppose that $z \sqsubseteq w$. We must show that $w = \bigvee \delta_{x,w,P}$ implies P. But $y \sqsubset z$, so we know that the equality $w = \bigvee \delta_{y,w,P}$ implies P. Now observe that $\bigvee \delta_{x,w,P} \sqsubseteq \bigvee \delta_{y,w,P}$, so if $w = \bigvee \delta_{x,w,P}$, then $w = \bigvee \delta_{y,w,P}$, finishing the proof. (ii) Assume $x \sqsubset y \sqsubseteq z$, let P be an arbitrary proposition in $\mathcal{V}$ and suppose that $z \sqsubseteq w$. We must show that $w = \bigvee \delta_{x,w,P}$ implies P. But $x \sqsubset y$ and $y \sqsubseteq w$, so this follows immediately. □

Proposition 6.2.12. *The following are equivalent for an element y of a $\mathcal{V}$-sup-lattice X:*
(i) the least element of X is strictly below y;
(ii) for every family $\alpha : I \to X$ with $I : \mathcal{V}$, if $y \sqsubseteq \bigvee \alpha$, then I is inhabited;
(iii) there exists some $x : X$ with $x \sqsubset y$.

Proof. Write $\perp$ for the least element of X. By Lemma 6.2.11 we have:

$$\perp \sqsubset y \iff \exists_{x:X}(\perp \sqsubseteq x \sqsubset y) \iff \exists_{x:X}(x \sqsubset y),$$

which proves the equivalence of (i) and (iii). It remains to prove that (i) and (ii) are equivalent. Suppose that $\perp \sqsubset y$ and let $\alpha : I \to X$ with $y \sqsubseteq \bigvee \alpha$. Using

$\bot \sqsubset y \sqsubseteq \bigvee \alpha$ and Lemma 6.2.11, we have $\bot \sqsubset \bigvee \alpha$. Hence, we only need to prove $\bigvee \alpha \sqsubseteq \bigvee \delta_{\bot,\bigvee \alpha,\exists i:I}$, but $\alpha_j \sqsubseteq \bigvee \delta_{\bot,\bigvee \alpha,\exists i:I}$ for every $j : I$, so this is true indeed. For the converse, assume that y satisfies (ii), suppose $z \sqsupseteq y$ and let $P : \mathcal{V}$ be a proposition such that $z = \bigvee \delta_{\bot,z,P}$. We must show that P holds. But notice that $y \sqsubseteq z = \bigvee \delta_{\bot,z,P} = \bigvee((p : P) \mapsto z)$, so P must be inhabited as y satisfies (ii). □

Item (ii) in Proposition 6.2.12 says exactly that y is a positive element in the sense of [Joh84, p. 98]. Observe that (ii) makes sense for any poset, not just $\mathcal{V}$-sup-lattices: we don't need to assume the existence of suprema to formulate condition (ii), because we can rephrase $y \sqsubseteq \bigvee \alpha$ as "for every $x : X$, if x is an upper bound of α and x is below any other upper bound of α, then $y \sqsubseteq x$". Similarly, the notion of being strictly below makes sense for any poset. What Proposition 6.2.12 shows is that strictly below generalises Johnstone's positivity from a *unary* relation to a *binary* one. Another binary generalisation of positivity in a different direction is that of a positivity relation in formal topology [Sam03; CS18; CV16]. For a formal topology S, one considers a binary relation $\ltimes$ between S and its powerclass. Then $a \ltimes S$ implies that a is positive [CS18, p. 764], while sets of the form $\{a \in S \mid a \ltimes U\}$ are thought of as formal closed subsets [CV16].

Looking to strengthen the notion of a nontrivial poset, we make the following definitions.

Definition 6.2.13 (Positivity; cf. [Joh84, p. 98]).

(i) An element of a $\delta_{\mathcal{V}}$-complete poset is *positive* if it satisfies Proposition 6.2.12(iii).

(ii) A $\delta_{\mathcal{V}}$-complete poset X is *positive* if we have specified $x, y : X$ with x strictly below y.

Example 6.2.14 (Nontriviality and positivity in $\Omega_{\mathcal{V}}$ and $\mathcal{P}_{\mathcal{V}}(X)$).

(i) Consider an element P of the $\delta_{\mathcal{V}}$-complete poset $\Omega_{\mathcal{V}}$. The pair $(\mathbf{0}_{\mathcal{V}}, P)$ witnesses nontriviality of $\Omega_{\mathcal{V}}$ if and only if $\neg\neg P$ holds, while it witnesses positivity if and only if P holds.

(ii) Say that a subset $A : \mathcal{P}_{\mathcal{V}}(X)$ is *nonempty* if the type $\Sigma_{x:X}(x \in A)$ is nonempty, and *inhabited* if this type is inhabited. The pair $(\emptyset, A)$ witnesses nontriviality of $\mathcal{P}_{\mathcal{V}}(X)$ if and only if A is nonempty, while it witnesses positivity if and only if A is inhabited.

We describe how the notion of strictly below relates to compactness and the way-below relation from domain theory.

Proposition 6.2.15. *If $x \sqsubseteq y$ are unequal elements of a $\mathcal{V}$-dcpo D and y is compact, then $x \sqsubset y$ without needing to assume excluded middle. In particular, a compact element x of a $\mathcal{V}$-dcpo with a least element $\bot$ is positive if and only if $x \neq \bot$.*

Proof. Suppose that $x \sqsubseteq y$ are unequal and that y is compact. We are to show that $x \sqsubset y$. So assume we have $z \sqsupseteq y$ and a proposition $P : \mathcal{V}$ such that $y \sqsubseteq z = \bigvee \delta_{x,z,P}$. By compactness of y, there exists $i : \mathbf{1} + P$ such that $y \sqsubseteq \delta_{x,z,P}(i)$ already. But i can't be equal to $\mathrm{inl}(\star)$, since $x \neq y$ is assumed. Hence, $i = \mathrm{inr}(p)$ and P must hold. □

Note that $x \sqsubset y$ does not imply $x \ll y$ in general, because with excluded middle,

$x \sqsubset y$ is simply the conjunction of $x \sqsubseteq y$ and $x \neq y$, which does not imply $x \ll y$ in general. Also, the conjunction of $x \ll y$ and $x \neq y$ does not imply $x \sqsubset y$, as far as we know.

We end this section by summarising why we consider strictly below to be a suitable notion in our constructive framework. First of all, $x \sqsubset y$ coincides with $(x \sqsubseteq y)\times(x \neq y)$ in the presence of excluded middle, so it is compatible with classical logic. Secondly, we've seen in Example 6.2.9 that strictly below works well in the poset of truth values and in powersets, yielding familiar constructive strengthenings. Thirdly, being strictly below generalises Johnstone's notion of positivity from a unary to a binary relation. And finally, as we will see shortly, the derived notion of positive poset is exactly what we need to derive Ω-Resizing$_{\mathcal{V}}$ rather than the weaker $\Omega_{\neg\neg}$-Resizing$_{\mathcal{V}}$ in Theorem 6.2.21.

6.2.3 Retract lemmas

We show that the type of propositions in $\mathcal{V}$ is a retract of any positive $\delta_{\mathcal{V}}$-complete poset and that the type of $\neg\neg$-stable propositions in $\mathcal{V}$ is a retract of any nontrivial $\delta_{\mathcal{V}}$-complete poset.

Definition 6.2.16 ($\Delta_{x,y}$)**.** For a nontrivial $\delta_{\mathcal{V}}$-complete poset $(X, \sqsubseteq, x, y)$, we define $\Delta_{x,y} : \Omega_{\mathcal{V}} \to X$ by the assignment $P \mapsto \bigvee \delta_{x,y,P}$.

We will often omit the subscripts in $\Delta_{x,y}$ when it is clear from the context. We extend the definition of local smallness (Definition 3.2.12) from $\mathcal{V}$-dcpos to $\delta_{\mathcal{V}}$-complete posets.

Definition 6.2.17 (Local smallness, $\sqsubseteq_{\mathcal{V}}$)**.** A $\delta_{\mathcal{V}}$-complete poset is *locally small* if its order has $\mathcal{V}$-small values, in which case we often denote the order with values in $\mathcal{V}$ by $\sqsubseteq_{\mathcal{V}}$.

Lemma 6.2.18. *A locally small $\delta_{\mathcal{V}}$-complete poset X is nontrivial, witnessed by elements $x \sqsubseteq y$, if and only if the composite $\Omega_{\mathcal{V}}^{\neg\neg} \hookrightarrow \Omega_{\mathcal{V}} \xrightarrow{\Delta_{x,y}} X$ is a section.*

Proof. Suppose first that $(X, \sqsubseteq, x, y)$ is nontrivial and locally small. We define

$$\begin{aligned} r : X &\to \Omega_{\mathcal{V}}^{\neg\neg} \\ z &\mapsto z \not\sqsubseteq_{\mathcal{V}} x. \end{aligned}$$

Note that negated propositions are $\neg\neg$-stable, so r is well-defined. Let $P : \mathcal{V}$ be an arbitrary $\neg\neg$-stable proposition. We want to show that $r(\Delta_{x,y}(P)) = P$. By propositional extensionality, establishing logical equivalence suffices. Suppose first that P holds. Then $\Delta_{x,y}(P) \equiv \bigvee \delta_{x,y,P} = y$, so $r(\Delta_{x,y}(P)) = r(y) \equiv (y \not\sqsubseteq_{\mathcal{V}} x)$ holds by antisymmetry and our assumptions that $x \sqsubseteq y$ and $x \neq y$. Conversely, assume that $r(\Delta_{x,y}(P))$ holds, i.e. that we have $\bigvee \delta_{x,y,P} \not\sqsubseteq_{\mathcal{V}} x$. Since P is $\neg\neg$-stable, it suffices to derive a contradiction from $\neg P$. So assume $\neg P$. Then $x = \bigvee \delta_{x,y,P}$, so $r(\Delta_{x,y}(P)) = r(x) \equiv x \not\sqsubseteq_{\mathcal{V}} x$, which is false by reflexivity.

For the converse, assume that $\Omega_{\mathcal{V}}^{\neg\neg} \hookrightarrow \Omega_{\mathcal{V}} \xrightarrow{\Delta_{x,y}} X$ has a retraction $r : \Omega_{\mathcal{V}}^{\neg\neg} \to X$.

Then $\mathbf{0}_{\mathcal{V}} = r(\Delta_{x,y}(\mathbf{0}_{\mathcal{V}})) = r(x)$ and $\mathbf{1}_{\mathcal{V}} = r(\Delta_{x,y}(\mathbf{1}_{\mathcal{V}})) = r(y)$, where we used that $\mathbf{0}_{\mathcal{V}}$ and $\mathbf{1}_{\mathcal{V}}$ are $\neg\neg$-stable. Since $\mathbf{0}_{\mathcal{V}} \neq \mathbf{1}_{\mathcal{V}}$, we get $x \neq y$, so $(X, \sqsubseteq, x, y)$ is nontrivial, as desired. □

The appearance of the double negation in the above lemma is due to the definition of nontriviality. If we instead assume a positive poset X, then we can exhibit all of $\Omega_{\mathcal{V}}$ as a retract of X.

Lemma 6.2.19. *A locally small $\delta_{\mathcal{V}}$-complete poset X is positive, witnessed by elements $x \sqsubset y$, if and only if for every $z \sqsupseteq y$, the map $\Delta_{x,z} : \Omega_{\mathcal{V}} \to X$ is a section.*

Proof. Suppose first that $(X, \sqsubseteq, x, y)$ is positive and locally small and let $z \sqsupseteq y$ be arbitrary. We define

$$\begin{aligned} r_z : X &\mapsto \Omega_{\mathcal{V}} \\ w &\mapsto z \sqsubseteq_{\mathcal{V}} w. \end{aligned}$$

Let $P : \mathcal{V}$ be arbitrary proposition. We want to show that $r_z(\Delta_{x,z}(P)) = P$. Because of propositional extensionality, it suffices to establish a logical equivalence between P and $r_z(\Delta_{x,z}(P))$. If P holds, then $\Delta_{x,z}(P) = z$, so $r_z(\Delta_{x,z}(P)) = r_z(z) \equiv (z \sqsubseteq_{\mathcal{V}} z)$ holds as well by reflexivity. Conversely, assume that $r_z(\Delta_{x,z}(P))$ holds, i.e. that we have $z \sqsubseteq_{\mathcal{V}} \bigvee \delta_{x,z,P}$. Since $\bigvee \delta_{x,z,P} \sqsubseteq z$ always holds, we get $z = \bigvee \delta_{x,z,P}$ by antisymmetry. But by assumption and Lemma 6.2.11, the element x is strictly below z, so P must hold.
For the converse, assume that for every $z \sqsupseteq y$, the map $\Delta_{x,z} : \Omega_{\mathcal{V}} \to X$ has a retraction $r_z : X \to \Omega_{\mathcal{V}}$. We must show that the equality $z = \Delta_{x,z}(P)$ implies P for every $z \sqsupseteq y$ and proposition $P : \mathcal{V}$. Assuming $z = \Delta_{x,z}(P)$, we have $\mathbf{1}_{\mathcal{V}} = r_z(\Delta_{x,z}(\mathbf{1}_{\mathcal{V}})) = r_z(z) = r_z(\Delta_{x,z}(P)) = P$, so P must hold indeed. Hence, $(X, \sqsubseteq, x, y)$ is positive, as desired. □

6.2.4 Small completeness with resizing

We present our main theorems here, which show that, constructively and predicatively, nontrivial $\delta_{\mathcal{V}}$-complete posets are necessarily large and necessarily lack decidable equality.

Definition 6.2.20 (Smallness)**.** A $\delta_{\mathcal{V}}$-complete poset is *small* if it is locally small and its carrier is $\mathcal{V}$-small.

Theorem 6.2.21.

(i) There is a nontrivial small $\delta_{\mathcal{V}}$-complete poset if and only if $\Omega_{\neg\neg}$-Resizing$_{\mathcal{V}}$ holds.

(ii) There is a positive small $\delta_{\mathcal{V}}$-complete poset if and only if Ω-Resizing$_{\mathcal{V}}$ holds.

Proof. (i) Suppose that $(X, \sqsubseteq, x, y)$ is a nontrivial small $\delta_{\mathcal{V}}$-complete poset. Using Lemma 6.2.18, we can exhibit $\Omega_{\mathcal{V}}^{\neg\neg}$ as a retract of X. But X is $\mathcal{V}$-small by assumption, so by Theorem 2.9.8 the type $\Omega_{\mathcal{V}}^{\neg\neg}$ is $\mathcal{V}$-small as well. For the converse, note that $(\Omega_{\mathcal{V}}^{\neg\neg}, \to, \mathbf{0}_{\mathcal{V}}, \mathbf{1}_{\mathcal{V}})$ is a nontrivial locally small $\mathcal{V}$-sup-lattice with $\bigvee \alpha$ given by $\neg\neg\exists_{i:I}\alpha_i$.

And if we assume $\Omega_{\neg\neg}$-Resizing$_{\mathcal{V}}$, then it is small. (ii) Suppose that $(X, \sqsubseteq, x, y)$ is a positive small poset. By Lemma 6.2.19, we can exhibit $\Omega_{\mathcal{V}}$ as a retract of X. But X is $\mathcal{V}$-small by assumption, so by Theorem 2.9.8 the type $\Omega_{\mathcal{V}}$ is $\mathcal{V}$-small as well. For the converse, note that $(\Omega_{\mathcal{V}}, \to, \mathbf{0}_{\mathcal{V}}, \mathbf{1}_{\mathcal{V}})$ is a positive locally small $\mathcal{V}$-sup-lattice. And if we assume Ω-Resizing$_{\mathcal{V}}$, then it is small. □

Lemma 6.2.22 ([Esc+, `TypeTopology.DiscreteAndSeparated`]).

(i) Types with decidable equality are closed under retracts.

(ii) Types with $\neg\neg$-stable equality are closed under retracts.

Proof. (i) Let $s : X \to Y$ be a section with retraction r, assume that Y has decidable equality and let $x, y : X$ be arbitrary. Then $s(x) = s(y)$ is decidable by assumption. If $s(x) = s(y)$, then $x = r(s(x)) = r(s(y)) = y$; and if $s(x) \neq s(y)$, then certainly $x \neq y$. Thus, $x = y$ is decidable, as desired. (ii) Using the same notation as before, the type $s(x) = s(y)$ is assumed to be $\neg\neg$-stable. But then

$$\neg\neg(x = y) \xrightarrow{\text{functoriality of } \neg\neg} \neg\neg(s(x) = s(y)) \xrightarrow{\neg\neg\text{-stability}} s(x) = s(y) \xrightarrow{\text{apply } r} x = y,$$

so $x = y$ is $\neg\neg$-stable, completing the proof. □

Example 6.2.23 (Types with $\neg\neg$-stable equality). The simple types $\mathbb{N}$, $\mathbb{N} \to \mathbb{N}$, $\mathbb{N} \to \mathbb{N} \to \mathbb{N}$, etc., see [Esc+, `TypeTopology.SimpleTypes`], and the type of Dedekind real numbers [Esc+, `Various.Dedekind`] all have $\neg\neg$-stable equality, as does the type $\Omega_{\mathcal{U}}^{\neg\neg}$ of $\neg\neg$-stable propositions in any universe $\mathcal{U}$.

Theorem 6.2.24. *There is a nontrivial locally small $\delta_{\mathcal{V}}$-complete poset with decidable equality if and only if weak excluded middle in $\mathcal{V}$ holds.*

Proof. Suppose that $(X, \sqsubseteq, x, y)$ is a nontrivial locally small $\delta_{\mathcal{V}}$-complete poset with decidable equality. Then by Lemmas 6.2.18 and 6.2.22, the type $\Omega_{\mathcal{V}}^{\neg\neg}$ must have decidable equality too. But negated propositions are $\neg\neg$-stable, so this yields weak excluded middle in $\mathcal{V}$. For the converse, note that $\left(\Omega_{\mathcal{V}}^{\neg\neg}, \to, \mathbf{0}_{\mathcal{V}}, \mathbf{1}_{\mathcal{V}}\right)$ is a nontrivial locally small $\mathcal{V}$-sup-lattice that has decidable equality if and only if weak excluded middle in $\mathcal{V}$ holds. □

Theorem 6.2.25. *The following are equivalent:*

(i) there is a positive locally small $\delta_{\mathcal{V}}$-complete poset with $\neg\neg$-stable equality;

(ii) there is a positive locally small $\delta_{\mathcal{V}}$-complete poset with decidable equality;

(iii) excluded middle in $\mathcal{V}$ holds.

Proof. Note that (ii) $\Rightarrow$ (i), so we are left to show that (iii) $\Rightarrow$ (ii) and that (i) $\Rightarrow$ (iii). For the first implication, note that $(\Omega_{\mathcal{V}}, \to, \mathbf{0}_{\mathcal{V}}, \mathbf{1}_{\mathcal{V}})$ is a positive locally small $\mathcal{V}$-sup-lattice that has decidable equality if and only if excluded middle in $\mathcal{V}$ holds. To see that (i) implies (iii), suppose that $(X, \sqsubseteq, x, y)$ is a positive locally small $\delta_{\mathcal{V}}$-complete poset with $\neg\neg$-stable equality. Then by Lemmas 6.2.19 and 6.2.22, the type $\Omega_{\mathcal{V}}$ must have $\neg\neg$-stable equality. But this implies that $\neg\neg P \to P$ for every proposition P in $\mathcal{V}$ which is equivalent to excluded middle in $\mathcal{V}$. ☐

In particular, Theorem 6.2.25(i) shows that, constructively, none of the types from Example 6.2.23 can be equipped with the structure of a positive $\delta_{\mathcal{V}}$-complete poset. Hence, we cannot expect the Dedekind reals to form a positive bounded complete poset.

Lattices, bounded complete posets and dcpos are necessarily large and necessarily lack decidable equality in our predicative constructive setting. More precisely:

Corollary 6.2.26.
(i) *There is a nontrivial small $\mathcal{V}$-sup-lattice (or $\mathcal{V}$-bounded complete poset or $\mathcal{V}$-dcpo) if and only if $\Omega_{\neg\neg}$-*Resizing$_{\mathcal{V}}$ *holds.*
(ii) *There is a positive small $\mathcal{V}$-sup-lattice (or $\mathcal{V}$-bounded complete poset or $\mathcal{V}$-dcpo) if and only if Ω-*Resizing$_{\mathcal{V}}$ *holds.*
(iii) *There is a nontrivial locally small $\mathcal{V}$-sup-lattice (or $\mathcal{V}$-bounded complete poset or $\mathcal{V}$-dcpo) with decidable equality if and only if weak excluded middle in $\mathcal{V}$ holds.*
(iv) *There is a positive locally small $\mathcal{V}$-sup-lattice (or $\mathcal{V}$-bounded complete poset or $\mathcal{V}$-dcpo) with decidable equality if and only if excluded middle in $\mathcal{V}$ holds.*

The above notions of non-triviality and positivity are data rather than property. Indeed, a nontrivial poset X is (by definition) equipped with two designated points $x, y : X$ such that $x \sqsubseteq y$ and $x \neq y$. It is natural to wonder if the propositionally truncated versions of these two notions yield the same conclusions. We show that this is indeed the case if we assume univalence. The need for the univalence assumption comes from the fact that smallness is a property precisely if univalence holds, as shown in Propositions 2.9.3 and 2.9.4.

Definition 6.2.27 (Nontrivial/positive in an unspecified way)**.** A poset X is *nontrivial in an unspecified way* if there exist some elements $x, y : X$ such that $x \sqsubseteq y$ and $x \neq y$, i.e. $\exists_{x,y:X}((x \sqsubseteq y) \times (x \neq y))$. Similarly, we can define when a poset is *positive in an unspecified way* by truncating the notion of positivity.

Theorem 6.2.28. *Suppose that the universes $\mathcal{V}$ and $\mathcal{V}^+$ are univalent.*
(i) *There is a small $\delta_{\mathcal{V}}$-complete poset that is nontrivial in an unspecified way if and only if $\Omega_{\neg\neg}$-*Resizing$_{\mathcal{V}}$ *holds.*
(ii) *There is a small $\delta_{\mathcal{V}}$-complete poset that is positive in an unspecified way if and only if Ω-*Resizing$_{\mathcal{V}}$ *holds.*

Proof. (i) Suppose that X is a $\delta_{\mathcal{V}}$-complete poset that is nontrivial in an unspecified way. By Proposition 2.9.3 and univalence of $\mathcal{V}$ and $\mathcal{V}^+$, the type "$\Omega_{\mathcal{V}}^{\neg\neg}$ is $\mathcal{V}$-small" is a proposition. By the universal property of the propositional truncation, in proving that the type $\Omega_{\mathcal{V}}^{\neg\neg}$ is $\mathcal{V}$-small we can therefore assume that are given points $x, y : X$ with $x \sqsubseteq y$ and $x \neq y$. The result then follows from Theorem 6.2.21(i). (ii) By reduction to Theorem 6.2.21(ii). ☐

Similarly, we can prove the following theorem by reduction to Theorems 6.2.24 and 6.2.25.

Theorem 6.2.29.

(i) There is a locally small $\delta_{\mathcal{V}}$-complete poset with decidable equality that is nontrivial in an unspecified way if and only if weak excluded middle in $\mathcal{V}$ holds.

(ii) There is a locally small $\delta_{\mathcal{V}}$-complete poset with decidable equality that is positive in an unspecified way if and only if excluded middle in $\mathcal{V}$ holds.

6.3 Maximal points and fixed points

As is well known, in impredicative mathematics, a poset has suprema of all subsets if and only if it has infima of all subsets. Perhaps counter-intuitively, this "duality" theorem can be proved predicatively. However, in the absence of impredicativity, it is not possible to fulfil its hypotheses when trying to apply it, because there are no nontrivial examples.

To explain this, we first have to make the statement of the duality theorem precise. A single universe formulation is "every $\mathcal{V}$-small $\mathcal{V}$-sup-lattice has all infima of families indexed by types in $\mathcal{V}$". The usual proof, adapted from subsets to families, shows that this formulation is predicatively provable, but in our predicative setting Theorem 6.2.21 tells us that there are no nontrivial examples to apply it to.

It is natural to wonder whether the single universe formulation can be generalised to *locally* small $\mathcal{V}$-sup-lattices (with large carriers), resulting in a predicatively useful result. However, one of the anonymous reviewers of our submission [dJE22a] pointed out that this generalisation is false and suggested the ordinals as a counterexample in a set-theoretic setting: it is a class with suprema for all subsets but has no greatest element. This led us to prove (Section 6.3.2) in our type-theoretic context that the locally small, but large type of ordinals in a universe $\mathcal{V}$ is a $\mathcal{V}$-sup-lattice (with no greatest element).

Similarly, consider a generalised formulation of Tarski's theorem [Tar55] that allows for multiple universes, i.e. we define *Tarski's-Theorem*$_{\mathcal{V},\mathcal{U},\mathcal{T}}$ as the assertion that every monotone endofunction on a $\mathcal{V}$-sup-lattice with carrier in a universe $\mathcal{U}$ and order taking values in a universe $\mathcal{T}$ has a greatest fixed point. Then Tarski's-Theorem$_{\mathcal{V},\mathcal{V},\mathcal{V}}$ corresponds to the original formulation and, moreover, is provable predicatively, but not useful predicatively because Theorem 6.2.21 shows that its hypotheses can only be fulfilled for trivial posets. On the other hand, Tarski's-Theorem$_{\mathcal{V},\mathcal{V}^+,\mathcal{V}}$ is provably false because the identity map on the $\mathcal{V}$-sup-lattice of ordinals in $\mathcal{V}$ is a counterexample. Analogous considerations could be made for a lemma due to Pataraia [Pat97; Esc03] saying that every dcpo has a greatest monotone inflationary endofunction.

6.3.1 A predicative counterexample

Because the type of ordinals in $\mathcal{V}$ is not $\mathcal{V}$-small even impredicatively, the above does not rule out the possibility that a $\mathcal{V}$-sup-lattice X has all $\mathcal{V}$-infima provided X is $\mathcal{V}$-small impredicatively. To address this, we present an example of a $\mathcal{V}$-sup-lattice that is $\mathcal{V}$-

small impredicatively, but predicatively does not necessarily have a maximal element. In particular, it need not have a greatest element or all $\mathcal{V}$-infima.

Fix a proposition $P_{\mathcal{U}}$ in a universe $\mathcal{U}$. We consider its lifting (in the sense of Section 3.4) with respect to a universe $\mathcal{V}$, i.e. we consider $\mathcal{L}_{\mathcal{V}}(P_{\mathcal{U}}) \equiv \Sigma_{Q:\Omega_{\mathcal{V}}}(Q \to P_{\mathcal{U}})$, just like in Example 4.9.4. This is a subtype of $\Omega_{\mathcal{V}}$ and it is closed under $\mathcal{V}$-suprema (in particular, it contains the least element).

Example 6.3.1.
(i) If $P_{\mathcal{U}} :\equiv \mathbf{0}_{\mathcal{U}}$, then $\mathcal{L}_{\mathcal{V}}(P_{\mathcal{U}}) \simeq \left(\Sigma_{Q:\Omega_{\mathcal{V}}}\neg Q\right) \simeq \left(\Sigma_{Q:\Omega_{\mathcal{V}}}(Q = \mathbf{0}_{\mathcal{V}})\right) \simeq \mathbf{1}$.
(ii) If $P_{\mathcal{U}} :\equiv \mathbf{1}_{\mathcal{U}}$, then $\mathcal{L}_{\mathcal{V}}(P_{\mathcal{U}}) \equiv \left(\Sigma_{Q:\Omega_{\mathcal{V}}}(Q \to \mathbf{1}_{\mathcal{U}})\right) \simeq \Omega_{\mathcal{V}}$.

What makes $\mathcal{L}_{\mathcal{V}}(P_{\mathcal{U}})$ useful is the following observation.

Lemma 6.3.2. *Suppose that the poset $\mathcal{L}_{\mathcal{V}}(P_{\mathcal{U}})$ has a maximal element $Q : \Omega_{\mathcal{V}}$. Then $P_{\mathcal{U}}$ is equivalent to Q, which is the greatest element of $\mathcal{L}_{\mathcal{V}}(P_{\mathcal{U}})$. In particular, $P_{\mathcal{U}}$ is $\mathcal{V}$-small. Conversely, if $P_{\mathcal{U}}$ is equivalent to a proposition $Q : \Omega_{\mathcal{V}}$, then Q is the greatest element of $\mathcal{L}_{\mathcal{V}}(P_{\mathcal{U}})$.*

Proof. Suppose that $\mathcal{L}_{\mathcal{V}}(P_{\mathcal{U}})$ has a maximal element $Q : \Omega_{\mathcal{V}}$. We wish to show that $Q \simeq P_{\mathcal{U}}$. By definition of $\mathcal{L}_{\mathcal{V}}(P_{\mathcal{U}})$, we already have that $Q \to P_{\mathcal{U}}$. So only the converse remains. Therefore suppose that $P_{\mathcal{U}}$ holds. Then, $\mathbf{1}_{\mathcal{V}}$ is an element of $\mathcal{L}_{\mathcal{V}}(P_{\mathcal{U}})$. Obviously $Q \to 1_{\mathcal{V}}$, but Q is maximal, so actually $Q = 1_{\mathcal{V}}$, that is, Q holds, as desired. Thus, $Q \simeq P_{\mathcal{U}}$. It is then straightforward to see that Q is actually the greatest element of $\mathcal{L}_{\mathcal{V}}(P_{\mathcal{U}})$, since $\mathcal{L}_{\mathcal{V}}(P_{\mathcal{U}}) \simeq \Sigma_{Q':\Omega_{\mathcal{V}}}(Q' \to Q)$. For the converse, assume that $P_{\mathcal{U}}$ is equivalent to a proposition $Q : \Omega_{\mathcal{V}}$. Then, as before, $\mathcal{L}_{\mathcal{V}}(P_{\mathcal{U}}) \simeq \Sigma_{Q':\Omega_{\mathcal{V}}}(Q' \to Q)$, which shows that Q is indeed the greatest element of $\mathcal{L}_{\mathcal{V}}(P_{\mathcal{U}})$. □

Corollary 6.3.3. *The $\mathcal{V}$-sup-lattice $\mathcal{L}_{\mathcal{V}}(P_{\mathcal{U}})$ has all $\mathcal{V}$-infima if and only if $P_{\mathcal{U}}$ is $\mathcal{V}$-small.*

Proof. Suppose first that $\mathcal{L}_{\mathcal{V}}(P_{\mathcal{U}})$ has all $\mathcal{V}$-infima. Then it must have an infimum for the empty family $\mathbf{0}_{\mathcal{V}} \to \mathcal{L}_{\mathcal{V}}(P_{\mathcal{U}})$. But this infimum must be the greatest element of $\mathcal{L}_{\mathcal{V}}(P_{\mathcal{U}})$. So by Lemma 6.3.2 the proposition $P_{\mathcal{U}}$ must be $\mathcal{V}$-small.
Conversely, suppose that $P_{\mathcal{U}}$ is equivalent to a proposition $Q : \mathcal{V}$. Then the infimum of a family $\alpha : I \to \mathcal{L}_{\mathcal{V}}(P_{\mathcal{U}})$ with $I : \mathcal{V}$ is given by $(Q \times \Pi_{i:I}\alpha_i) : \mathcal{V}$. □

In [dJE21a] we used Lemma 6.3.2 to conclude that a version of Zorn's Lemma that says that every pointed dcpo has a maximal element is predicatively unavailable, as $\mathcal{L}_{\mathcal{V}}(P_{\mathcal{U}})$ is a pointed $\mathcal{V}$-dcpo, but has a maximal element if and only if $P_{\mathcal{U}}$ is $\mathcal{V}$-small. But, as in our above discussion of the duality theorem and Tarski's Theorem, we must pay attention to the universes here. Zorn's Lemma restricted to $\mathcal{V}$-small $\mathcal{V}$-sup-lattices is, assuming excluded middle [Bel97], equivalent to the axiom of choice, as usual. Disregarding its constructive status for a moment, the predicative issue is that there are no nontrivial $\mathcal{V}$-small $\mathcal{V}$-sup-lattices (Theorem 6.2.21). But the generalisation of Zorn's Lemma to *locally* small $\mathcal{V}$-sup-lattices is false (even if we assume the axiom of choice and hence, excluded middle), because the $\mathcal{V}$-sup-lattice of ordinals in $\mathcal{V}$, having no maximal element, is a counterexample.

6.3.2 Small suprema of small ordinals

We now show that the ordinal $\mathrm{Ord}_{\mathcal{V}}$ of ordinals in a fixed *univalent* universe $\mathcal{V}$ has suprema for all families indexed by types in $\mathcal{V}$ and that it has no maximal element. The latter is implied by [Uni13, Lemma 10.3.21], but we were not able to find a proof of the former in the literature: Theorem 9 of [KNX21] only proves $\mathrm{Ord}_{\mathcal{V}}$ to have joins of increasing sequences, while [Uni13, Lemma 10.3.22] shows that every family indexed by a type in $\mathcal{V}$ has some upper bound, but does not prove it to be the least (although least upper bounds are required for [Uni13, Exercise 10.17(ii)]). We present two proofs: one based on [Uni13, Lemma 10.3.22] using small set quotients and an alternative one using small images.

Following [Uni13, Section 10.3], we define an ordinal to be a type equipped with a proposition-valued, transitive, extensional and (inductive) well-founded relation. In [Uni13] the underlying type of an ordinal is required to be a set, but this actually follows from the other axioms, see [Esc+, `Ordinals.Type`]. The type of ordinals, denoted by $\mathrm{Ord}_{\mathcal{V}}$, in a given *univalent* universe $\mathcal{V}$ can itself be equipped with such a relation [Uni13, Theorem 10.3.20] and thus is an ordinal again. However, it is not an ordinal in $\mathcal{V}$, but rather in the next universe $\mathcal{V}^+$, and this is necessary, because it is contradictory for $\mathrm{Ord}_{\mathcal{V}}$ to be isomorphic to an ordinal in $\mathcal{V}$, see [BCDE20]. Before we can prove that $\mathrm{Ord}_{\mathcal{V}}$ has $\mathcal{V}$-suprema, we need to recall a few facts. The well-order on $\mathrm{Ord}_{\mathcal{V}}$ is given by: $\alpha \prec \beta$ if and only if there exists a (necessarily) unique $y : \beta$ such that α and $\beta \downarrow y$ are isomorphic ordinals. Here $\beta \downarrow y$ denotes the ordinal of elements $b : \beta$ satisfying $b \prec y$.

Lemma 6.3.4 ([Esc+, `Ordinals.OrdinalOfOrdinals`]). *For every two points x and y of an ordinal α, we have $x \prec y$ in α if and only if $\alpha \downarrow x \prec \alpha \downarrow y$ as ordinals.*

Definition 6.3.5 (Simulation; [Uni13, Section 10.3]). A *simulation* between two ordinals α and β is a map $f : \alpha \to \beta$ satisfying the following conditions:

(i) for every $x, y : \alpha$, if $x \prec y$, then $f(x) \prec f(y)$;

(ii) for every $x : \alpha$ and $y : \beta$, if $y \prec f(x)$, then there exists a (necessarily unique) $x' : \alpha$ such that $x' \prec x$ and $f(x') = y$.

Lemma 6.3.6 ([Esc+, `Ordinals.OrdinalOfOrdinals`]). *For ordinals α and β, the following are equivalent:*

(i) there exists a (necessarily unique) simulation from α to β;

(ii) for every ordinal γ, if $\gamma \prec \alpha$, then $\gamma \prec \beta$.

We write $\alpha \preceq \beta$ if the equivalent conditions above hold.

Recall from Definition 2.11.23 what it means to have small set quotients. If these are available, then the type of ordinals has all small suprema.

Theorem 6.3.7 (Extending [Uni13, Lemma 10.3.22]). *Assuming small set quotients, the large ordinal $\mathrm{Ord}_{\mathcal{V}}$ has suprema of families indexed by types in $\mathcal{V}$.*

Proof. Given $\alpha : I \to \mathrm{Ord}_{\mathcal{V}}$, define $\hat{\alpha}$ as the quotient of $\Sigma_{i:I}\, \alpha_i$ by the $\mathcal{V}$-valued equivalence relation $\approx$ where $(i, x) \approx (j, y)$ if and only if $\alpha_i \downarrow x$ and $\alpha_j \downarrow y$ are isomorphic as ordinals. By our assumption, the quotient $\hat{\alpha}$ lives in $\mathcal{V}$. Next, we use [Uni13, Lemma 10.3.22] which tells us that $(\hat{\alpha}, \prec)$ with

$$[(i,x)] \prec [(j,y)] :\equiv (\alpha_i \downarrow x) \prec (\alpha_j \downarrow y).$$

is an ordinal that is an upper bound of α. So we show that $\hat{\alpha}$ is a lower bound of upper bounds of α. To this end, suppose that $\beta : \mathrm{Ord}_{\mathcal{V}}$ is such that $\alpha_i \preceq \beta$ for every $i : I$. In light of Lemma 6.3.6, this assumption yields two things:

(1) for every $i : I$ and $x : \alpha_i$ there exists a unique $b_i^x : \beta$ such that $\alpha_i \downarrow x = \beta \downarrow b_i^x$;
(2) for every $i : I$, a simulation $f_i : \alpha_i \to \beta$ such that for every $x : \alpha_i$, we have $f_i(x) = b_i^x$.

We are to prove that $\hat{\alpha} \preceq \beta$. We start by defining

$$\begin{aligned} f : (\Sigma_{i:I}\, \alpha_i) &\to \beta \\ (i,x) &\mapsto b_i^x. \end{aligned}$$

Observe that f respects $\approx$, for if $(i,x) \approx (j,y)$, then by univalence,

$$(\beta \downarrow b_i^x) = (\alpha_i \downarrow x) = (\alpha_j \downarrow y) = (\beta \downarrow b_j^y),$$

so $b_i^x = b_j^y$ by uniqueness of b_i^x. Thus, f induces a map $\hat{f} : \hat{\alpha} \to \beta$ satisfying the equality $\hat{f}([(i,x)]) = f(i,x)$ for every $(i,x) : \Sigma_{j:J}\, \alpha_j$.
It remains to prove that $\hat{f}$ is a simulation. Because the defining properties of a simulation are propositions, we can use set quotient induction and it suffices to prove the following two things:

(I) if $\alpha_i \downarrow x \prec \alpha_j \downarrow y$, then $b_i^x \prec b_j^y$;
(II) if $b \prec b_i^x$, then there exists $j : I$ and $y : \alpha_j$ such that $\alpha_i \downarrow y \prec \alpha_j \downarrow x$ and $b_j^y = b$.

For (I), observe that if $\alpha_i \downarrow x \prec \alpha_j \downarrow y$, then $\beta \downarrow b_i^x \prec \beta \downarrow b_j^y$, from which $b_i^x \prec b_j^y$ follows using Lemma 6.3.4. For (II) suppose that $b \prec b_i^x$. Because f_i (see item (2) above) is a simulation, there exists $y : \alpha_i$ with $y \prec x$ and $f_i(y) = b$. By Lemma 6.3.4, we get $\alpha_i \downarrow y \prec \alpha_i \downarrow x$. Moreover, $b_i^y = f_i(y) = b$, finishing the proof of (II). □

In Section 2.11.4 we saw that set replacement is equivalent to the existence of small set quotients, so the following result immediately follows from the theorem above. But the point is that an alternative construction without set quotients is available, if set replacement is assumed.

Theorem 6.3.8. *Assuming set replacement, the large ordinal* $\mathrm{Ord}_{\mathcal{V}}$ *has suprema of families indexed by types in* $\mathcal{V}$*.*

Proof. Given $\alpha : I \to \mathrm{Ord}_{\mathcal{V}}$, consider the image of the map $e : \Sigma_{i:I}\, \alpha_i \to \mathrm{Ord}_{\mathcal{V}}$ given by $e(i,x) :\equiv \alpha_i \downarrow x$. The image of e is conveniently equivalent to the type $\Sigma_{\gamma:\mathrm{Ord}_{\mathcal{V}}} \exists_{i:I}\, \gamma \prec \alpha_i$, i.e. the type of ordinals that are initial segments of some α_i. One can prove that $\mathrm{im}(e)$ with the induced order from $\mathrm{Ord}_{\mathcal{V}}$ is again a well-order and that for every $i : I$, the canonical map $\alpha_i \to \mathrm{im}(e)$ is a simulation. Moreover, if β is an ordinal such that $\alpha_i \preceq \beta$ for every $i : I$, then for every $i : I$ and every $x : \alpha_i$ there exists a unique $b_i^x : \beta$ such that $\alpha_i \downarrow x = \beta \downarrow b_i^x$. Now observe that for every $\gamma : \mathrm{Ord}_{\mathcal{V}}$, the map $\left(\Sigma_{i:I}\Sigma_{x:\alpha_i}(\alpha_i \downarrow x = \gamma)\right) \to \beta$ defined by the assignment $(i, x, p) \mapsto b_i^x$ is a constant function to a set. Hence, by Theorem 2.6.9, this map factors through the propositional truncation $\exists_{i:I}\Sigma_{x:\alpha_i}(\alpha \downarrow x = \gamma)$. This yields a map $\mathrm{im}(e) \to \beta$ which can be proved to be a simulation, as desired. Finally, we use set replacement and the fact that $\mathrm{Ord}_{\mathcal{V}}$ is locally $\mathcal{V}$-small (by univalence) to get an ordinal in $\mathcal{V}$ equivalent to $\mathrm{im}(e)$, finishing the proof. □

6.4 Families and subsets

In traditional impredicative foundations, completeness of posets is usually formulated using subsets. For instance, dcpos are defined as posets D such that every directed subset of D has a supremum in D. Example 6.2.3 are all formulated using small families instead of subsets. While subsets are primitive in set theory, families are primitive in type theory, so this could be an argument for using families above. However, that still leaves the natural question of how the family-based definitions compare to the usual subset-based definitions, especially in our predicative setting, unanswered. This section addresses this question. We first study the relation between subsets and families predicatively and then clarify our definitions in the presence of impredicativity. In our answers we will consider sup-lattices, but similar arguments could be made for posets with other sorts of completeness, such as dcpos.

We first show that simply asking for completeness with respect to all subsets is not satisfactory from a predicative viewpoint. In fact, we will now see that even asking for completeness with respect to all elements of $\mathcal{P}_{\mathcal{T}}(X)$ for some fixed universe $\mathcal{T}$ is problematic from a predicative standpoint, where we recall from Definition 2.7.5 that we refer to the elements of $\mathcal{P}_{\mathcal{T}}(X) \equiv (X \to \Omega_{\mathcal{T}})$ as $\mathcal{T}$-valued subsets of X.

Theorem 6.4.1. *Let $\mathcal{U}$ and $\mathcal{V}$ be universes, fix a proposition $P_{\mathcal{U}} : \mathcal{U}$ and recall $\mathcal{L}_{\mathcal{V}}(P_{\mathcal{U}})$ defined in Section 6.3.1, which has $\mathcal{V}$-suprema. If $\mathcal{L}_{\mathcal{V}}(P_{\mathcal{U}})$ has suprema for all $\mathcal{T}$-valued subsets, then $P_{\mathcal{U}}$ is $\mathcal{V}$-small independently of the choice of the type universe $\mathcal{T}$.*

Proof. Let $\mathcal{T}$ be a type universe and consider the subset S of $\mathcal{L}_{\mathcal{V}}(P_{\mathcal{U}})$ given by $Q \mapsto \mathbf{1}_{\mathcal{T}}$. Note that S has a supremum in $\mathcal{L}_{\mathcal{V}}(P_{\mathcal{U}})$ if and only if $\mathcal{L}_{\mathcal{V}}(P_{\mathcal{U}})$ has a greatest element, but by Lemma 6.3.2, the latter is equivalent to $P_{\mathcal{U}}$ being $\mathcal{V}$-small. □

The proof above illustrates that if we have a subset $S : \mathcal{P}_{\mathcal{T}}(X)$, then there is no reason why the total space $\Sigma_{x:X}(x \in S)$ (recall Definition 4.2.8) should be $\mathcal{T}$-small. In fact, for $S(x) :\equiv \mathbf{1}_{\mathcal{T}}$ as above, the latter is equivalent to asking that X is $\mathcal{T}$-small.

In an attempt to solve the problem described in Theorem 6.4.1, we look to impose size restrictions on the total space of a subset. There are two natural such restrictions and they are reminiscent of Bishop and Kuratowski finite subsets.

Definition 6.4.2 ($\mathcal{V}$-small and $\mathcal{V}$-covered subsets)**.** An element $S : \mathcal{P}_{\mathcal{T}}(X)$ is

(i) $\mathcal{V}$-*small* if its total space is $\mathcal{V}$-small, and
(ii) $\mathcal{V}$-*covered* if we have $I : \mathcal{V}$ with a surjection $e : I \twoheadrightarrow \mathbb{T}(S)$.

Observe that every $\mathcal{V}$-small subset is $\mathcal{V}$-covered, because every equivalence is a surjection. But the converse does not hold: We can emulate the well-known argument used to show that, constructively, Kuratowski finiteness does not necessarily imply Bishop finiteness to show that, predicatively, being $\mathcal{V}$-covered does not necessarily imply being $\mathcal{V}$-small.

Proposition 6.4.3. *For every two universes $\mathcal{U}$ and $\mathcal{V}$, if every $\mathcal{V}$-covered element of $\mathcal{P}_{\mathcal{U}}(\Omega_{\mathcal{U}})$ is $\mathcal{V}$-small, then* Propositional-Resizing$_{\mathcal{U},\mathcal{V}}$ *holds.*

Proof. Suppose that every $\mathcal{V}$-covered $\mathcal{U}$-valued subset of $\Omega_{\mathcal{U}}$ is $\mathcal{V}$-small and let $P : \mathcal{U}$ be an arbitrary proposition. Consider the subset $S_P : \Omega_{\mathcal{U}} \to \Omega_{\mathcal{U}}$ given by $S_P(Q) :\equiv (Q = P) \vee (Q = \mathbf{1}_{\mathcal{U}})$. Notice that this is $\mathcal{V}$-covered as witnessed by

$$\begin{aligned}(\mathbf{1}_{\mathcal{V}} + \mathbf{1}_{\mathcal{V}}) &\twoheadrightarrow \mathbb{T}(S_P)\\ \mathrm{inl}(\star) &\mapsto (P\,, |\mathrm{inl}(\mathrm{refl})|)\\ \mathrm{inr}(\star) &\mapsto (\mathbf{1}_{\mathcal{U}}, |\mathrm{inr}(\mathrm{refl})|),\end{aligned}$$

so by assumption $\mathbb{T}(S_P)$ is $\mathcal{V}$-small. But observe that P holds if and only if $\mathbb{T}(S_P)$ is a subsingleton, but the latter type is $\mathcal{V}$-small by assumption, hence so is P. □

In the case where we restrict our attention to a single universe $\mathcal{V}$ and a locally $\mathcal{V}$-small set X, the two notions coincide if and only if we have set replacement for maps into X with $\mathcal{V}$-small domain.

Proposition 6.4.4. *If X is a locally $\mathcal{V}$-small set, then every $\mathcal{V}$-covered element of $\mathcal{P}_{\mathcal{V}}(X)$ is $\mathcal{V}$-small if and only if the image of any map into X with a $\mathcal{V}$-small domain is $\mathcal{V}$-small.*

Proof. Suppose first that every $\mathcal{V}$-covered subset $S : X \to \Omega_{\mathcal{V}}$ is $\mathcal{V}$-small and let $f : I \to X$ be a map such that I is $\mathcal{V}$-small. Without loss of generality, we may assume that $I : \mathcal{V}$, because we can always precompose f with the equivalence witnessing that I is $\mathcal{V}$-small. Now consider the subset $S : X \to \Omega_{\mathcal{V}}$ given by $S(x) :\equiv \exists_{i:I}(f(i) =_{\mathcal{V}} x)$, where $=_{\mathcal{V}}$ has values in $\mathcal{V}$ and is provided by our assumption that X is locally $\mathcal{V}$-small. Then S is $\mathcal{V}$-covered, because we have $I \twoheadrightarrow \mathrm{im}(f) \simeq \mathbb{T}(S)$, where the first map is the corestriction of f. So by assumption $\mathbb{T}(S)$ is $\mathcal{V}$-small, which means that $\mathrm{im}(f)$ must be $\mathcal{V}$-small too.
Conversely, assume the set replacement principle and let $S : X \to \Omega_{\mathcal{V}}$ be $\mathcal{V}$-covered by $e : I \twoheadrightarrow \mathbb{T}(S)$. Define the subset $S' : X \to \Omega_{\mathcal{V}}$ by $S'(x) :\equiv \exists_{i:I}(x =_{\mathcal{V}} \mathrm{pr}_1(e_i))$. By the assumed set replacement principle for X, the subset S' is $\mathcal{V}$-small because of the equivalence $\mathbb{T}(S') \simeq \mathrm{im}(\mathrm{pr}_1 \circ e)$. Finally, it follows from the surjectivity of e that S and S' are equal as subsets, and therefore that $\mathbb{T}(S) \simeq \mathbb{T}(S')$. Hence, S is a $\mathcal{V}$-small subset, as desired. □

So, predicatively, and in the absence of a set replacement principle, the notion of a $\mathcal{V}$-small subset is strictly stronger than that of a $\mathcal{V}$-covered subset. Hence, in this setting, having suprema for all $\mathcal{V}$-small subsets is strictly weaker than having suprema for all $\mathcal{V}$-covered subsets. Meanwhile, Corollary 6.4.6 will imply that there are plenty of examples of posets with suprema for all $\mathcal{V}$-covered subsets, even predicatively. So we prefer the stronger, but predicatively reasonable requirement of asking for suprema of all $\mathcal{V}$-covered subsets.

Form a practical viewpoint, $\mathcal{V}$-covered subsets also give us an easy handle on examples like the following: Let X be a poset with suprema for all (directed) $\mathcal{U}_0$-covered subsets. Then the least fixed point of a Scott continuous endofunction f on X can be computed as the supremum of the subset $\{\bot, f(\bot), f^2(\bot), \dots\}$, which is covered by $\mathbb{N}$. But it is not clear that this subset is $\mathcal{U}_0$-small, at least not in the absence of set replacement.

Our preference for $\mathcal{V}$-covered subsets over $\mathcal{V}$-small subsets also makes it clear why we do not impose an injectivity condition on families, because for every type $X : \mathcal{U}$ there is an equivalence between embeddings $I \hookrightarrow X$ with $I : \mathcal{V}$ and $(\mathcal{U} \sqcup \mathcal{V})$-valued subsets of X whose total spaces are $\mathcal{V}$-small, cf. [Esc+, `Slice.Slice`].

Theorem 6.4.5. *For $X : \mathcal{U}$ and any universe $\mathcal{V}$ we have an equivalence between $\mathcal{V}$-covered $(\mathcal{U} \sqcup \mathcal{V})$-valued subsets of X and families $I \to X$ with $I : \mathcal{V}$.*

Proof. The forward map φ is given by $(S, I, e) \mapsto (I, \mathrm{pr}_1 \circ e)$. In the other direction, we define ψ by mapping (I, α) to the triple (S, I, e) where S is the subset of X given by $S(x) :\equiv \exists_{i:I}\,(x = \alpha(i))$ and $e : I \twoheadrightarrow \mathbb{T}(S)$ is defined as $e(i) :\equiv (\alpha(i), |(i, \mathrm{refl})|)$. The composite $\varphi \circ \psi$ is easily seen to be equal to the identity. To show that $\psi \circ \varphi$ equals the identity, we need the following intermediate result, which is proved using function extensionality and path induction.

Claim. Let $S, S' : X \to \Omega_{\mathcal{U} \sqcup \mathcal{V}}$, $e : I \to \mathbb{T}(S)$ and $e' : I \to \mathbb{T}(S')$. If $S = S'$ and $\mathrm{pr}_1 \circ e \sim \mathrm{pr}_1 \circ e'$, then $(S, e) = (S', e')$.

The result follows from the claim using function and propositional extensionality. □

Corollary 6.4.6. *A poset with carrier in a universe $\mathcal{U}$ has suprema for all $\mathcal{V}$-covered $(\mathcal{U} \sqcup \mathcal{V})$-valued subsets if and only if it has suprema for all families indexed by types in $\mathcal{V}$.*

Proof. This is because the supremum of a $\mathcal{V}$-covered subset equals the supremum of the corresponding family and vice versa by inspecting the proof of Theorem 6.4.5. □

We conclude by comparing our family-based approach to the subset-based approach in the presence of impredicativity.

Theorem 6.4.7. *Assuming Ω-Resizing$_{\mathcal{T},\mathcal{U}_0}$ for every universe $\mathcal{T}$. Then the following are equivalent for a poset with carrier in a universe $\mathcal{U}$:*

(i) the poset has suprema for all subsets;
(ii) the poset has suprema for all $\mathcal{U}$-covered subsets;
(iii) the poset has suprema for all $\mathcal{U}$-small subsets;
(iv) the poset has suprema for all families indexed by types in $\mathcal{U}$.

Proof. Clearly (i) ⇒ (ii) ⇒ (iii). We show that (iii) implies (i), which proves the equivalence of (i)–(iii). Assume that a poset X has suprema for all $\mathcal{U}$-small subsets and let $S : X \to \Omega_{\mathcal{T}}$ be any subset of X. Using Ω-Resizing$_{\mathcal{T},\mathcal{U}_0}$, the total space $\mathbb{T}(S)$ is $\mathcal{U}$-small. So X has a supremum for S by assumption, as desired. Finally, (ii) and (iv) are equivalent in the presence of Ω-Resizing$_{\mathcal{T},\mathcal{U}_0}$ by Corollary 6.4.6. □

If condition (iv) of Theorem 6.4.7 holds, then the poset has suprema for all families indexed by types in $\mathcal{V}$ provided that $\mathcal{V} \sqcup \mathcal{U} \equiv \mathcal{U}$. Typically, in the examples in our account of domain theory for instance, $\mathcal{U} :\equiv \mathcal{U}_1$ and $\mathcal{V} :\equiv \mathcal{U}_0$, so that $\mathcal{V} \sqcup \mathcal{U} \equiv \mathcal{U}$ holds. Thus, our $\mathcal{V}$-families-based approach generalises the traditional subset-based approach.

6.5 Notes

This chapter is based on an extended and revised version [dJE22a, Sections 4–6] of our paper [dJE21b]. We would like to thank the anonymous reviewers of [dJE22a] for their valuable and complementary suggestions. We are particularly grateful to the reviewer who pointed out that one of our results can be strengthened to Theorem 2.9.8 and for their insights and questions on Sections 6.3 and 6.4 that have considerably improved the exposition.

Chapter 7

Formalisation

This research started with formalising the Scott model of PCF in Coq [Coq] using the UniMath [VAG+] library. The UniMath project was started in 2014 by merging repositories from Vladimir Voevodsky, Benedikt Ahrens and Daniel Grayson. The current UniMath Coordinating Committee members are: Benedikt Ahrens, Daniel Grayson, Ralph Matthes and Niels van der Weide and the library has accepted contributions[1] from over 60 people at the time of writing.

For historical reasons [Voe15], the `Type-in-Type` flag is enabled in UniMath, so that it is not possible to have Coq automatically check the universes for us. Since we were interested in developing domain theory predicatively, having the proof assistant carefully track universes was an important feature. Hence, we decided to continue our formalisation efforts in Agda [NDCA+] using Martín Escardó's (and collaborators') TypeTopology [Esc+] development which explicitly keeps track of universes. This is the reason some parts are formalised both in Coq/UniMath and in Agda/TypeTopology. But because we did not wish to duplicate all our efforts, some parts are only formalised in Coq/UniMath.

Formalising our efforts has helped to experiment with and has structured and guided our development of domain theory as set out in this thesis. Moreover [Har20] has shown that the Agda formalisation [dJon22a] can be taken as a starting point for a further formal development of domain theory in univalent foundations. Since then, the code base has been extended and improved considerably, hopefully making our formalisation more convenient to work with.

Both the Coq/UniMath and Agda/TypeTopology proofs and their renderings in HTML (for presentation and reading) are archived by the University of Birmingham at `doi:10.25500/edata.bham.00000912`.

[1] `https://github.com/UniMath/UniMath/graphs/contributors`

7.1 Overview

We give a comprehensive overview per chapter of what is and what isn't formalised.

Chapter 2 All the background material up to and including Section 2.10 is fairly well-known and all formalised between [Esc+] and [Esc19b], except for Theorem 2.9.8 because we only recently learned that this stronger result was possible, see Remark 2.9.9. The weaker result for sections that are embeddings is formalised however and is sufficient for our applications.

Section 2.11 on set quotients, set replacement and propositional truncations is formalised in Agda/TypeTopology, see [Esc18; dJE21c; dJon22d; dJon22b].

Section 2.12 on indexed W-types is formalised in Coq/UniMath, see [dJon19, `MoreFoundations.Wtypes`], as it was aimed at the application to PCF.

Chapter 3 All of Chapter 3 is formalised in Agda/TypeTopology, see [dJon22a], except for

- products of dcpos (although this was included in [Har20]),
- Propositions 3.4.1 and 3.4.6, and
- Theorem 3.4.19 (but the similar Theorems 3.4.17 and 3.4.22 are formalised).

Chapter 4 All of Chapter 4 is formalised in Agda/TypeTopology, see [dJon22a], with Example 4.9.4, and Lemmas 4.13.11 and 4.13.12 as the only exceptions.

Chapter 5 Section 5.1 is fully formalised in Agda/TypeTopology, see the dedicated file [dJon22a, `Bilimits.Dinfinity`]. Regarding Section 5.2, we would like to mention three formalisations.

The Scott model of PCF, including soundness and computational adequacy, is fully formalised in Coq/UniMath, see [dJon19, `Partiality.PCF`]. This also includes the general Proposition 5.2.24 in [dJon19, `MoreFoundations.ClosureOfHrel`], and the logical equivalence of Theorem 5.2.25, but the application of Proposition 5.2.24 to obtain decidability of $\triangleright^k$ is not formalised.

To check the predicative validity and the universes involved, which is not possible in Coq/UniMath because it uses `Type-in-Type` (as mentioned above), we also formalised the syntax of PCF and the definition of the Scott model of PCF in Agda/TypeTopology, see [dJon22a, `ScottModelOfPCF.ScottModelOfPCF`].

This Agda development was extended by Brendan Hart [Har20] to a proof of soundness and computational adequacy for PCF with variables and λ-abstraction for a final year MSci project supervised by Martín Escardó and myself.

Chapter 6 Only the results of Section 6.3.2 on suprema of ordinals have been formalised in Agda/TypeTopology, see [dJE22b].

7.2 Future work

It would be desirable to expand [dJon22a] with a formalisation of products and Lemmas 4.13.11 and 4.13.12. Products are not included because they were not needed to develop the applications in Chapter 5, while the lemmas are missing due to a lack of time. Including products could potentially be achieved by merging [Har20]. The other results are less pressing to have formalised, because they do not obstruct a further computer-verified development of domain theory, for example because they are results showing that some statement implies a constructive or predicative taboo.

7.3 Statistics

To give the reader an impression of the (relative) sizes of our formalisations of domain theory, Table 7.3.1 and Table 7.3.2 show the number of lines in respectively [dJon19] and [dJon22a]. Notice that the files listed in these tables depend on auxiliary files that develop univalent foundations (just as Chapters 3 to 5 depend on Chapter 2), but that these auxiliary files are not included in the statistics.

File	Number of lines
`DCPO.v`	637
`LiftMonad.v`	151
`PartialElements.v`	369
`PCF.v`	769
	1953

Table 7.3.1: Number of lines (including comments and blank lines) per file in our COQ/UNIMATH formalisation [dJon19]. It should be noted that the file `PCF.v` in Table 7.3.1 includes the soundness and computational adequacy of the Scott model of PCF.

File	Number of lines
Basics	
`Dcpo.lagda`	303
`Exponential.lagda`	346
`LeastFixedPoint.lagda`	309
`Miscelanea.lagda`	668
`Pointed.lagda`	341
`SupComplete.lagda`	294
`WayBelow.lagda`	289
	2550

BasesAndContinuity	
`Bases.lagda`	721
`Continuity.lagda`	502
`ContinuityDiscussion.lagda`	375
`IndCompletion.lagda`	379
`StepFunctions.lagda`	509
	3716
Bilimits	
`Dinfinity.lagda`	962
`Directed.lagda`	1229
`Sequential.lagda`	495
	2686
Examples	
`IdlDyadics.lagda`	80
`Omega.lagda`	219
`Powerset.lagda`	216
	515
IdealCompletion	
`IdealCompletion.lagda`	198
`Properties.lagda`	528
`Retracts.lagda`	472
	1198
Lifting	
`LiftingDcpo.lagda`	466
`LiftingSet.lagda`	395
`LiftingSetAlgebraic.lagda`	208
	1069
ScottModelOfPCF	
`ScottModelOfPCF.lagda`	71
`PCF.lagda`	118
`PCFCombinators.lagda`	474
	663
All components combined	11167

Table 7.3.2: Number of lines (including comments and blank lines) per component of our Agda/TypeTopology formalisation [dJon22a]

CHAPTER 8

Conclusion

We provide a summary of our contributions and our approach to developing domain theory in constructive and predicative univalent foundations. Furthermore, we briefly describe various directions for future research.

8.1 Summary

We have demonstrated how constructive and predicative univalent foundations provides an adequate and sophisticated setting to develop domain theory. Since higher inductive types can be seen as specific instances of resizing principles, it is noteworthy that the only higher inductive type needed for our purposes is the propositional truncation.

Instead of working with information systems, abstract bases or formal topologies, and approximable relations, we studied directed complete posets and Scott continuous directly, using type universes and type equivalences to deal with size issues in the absence of propositional resizing axioms. Seeing a poset as a category in the usual way, we can say that dcpos are large, but locally small, and have small filtered colimits. By carefully keeping track of universe parameters, we showed that complex constructions of dcpos, such as Scott's D_∞ model of the untyped λ-calculus, which involves countable infinite iterations of exponentials, are predicatively possible. We further illustrated our domain-theoretic development by giving a predicative and constructive account of the soundness and computational adequacy of the Scott model of PCF. In particular, this illustrated the use of the Escardó–Knapp lifting monad.

Taking inspiration from work in category theory by Johnstone and Joyal, we gave predicatively adequate definitions of continuous and algebraic dcpos, and discussed issues related to the absence of the axiom of choice. We also presented predicative adaptations of the notions of a basis and the rounded ideal completion. The theory was accompanied by several examples: we described small compact bases for the lifting and the powerset, and considered the rounded ideal completion of the dyadics.

The fact that nontrivial dcpos have large carriers is in fact unavoidable and characteristic of our predicative setting, as we explained in a complementary chapter on the constructive and predicative limitations of univalent foundations. We proved no-go theorems regarding both constructivity and predicativity for a general class of posets that includes dcpos, bounded complete posets, sup-lattices and frames. In particular, we showed that locally small nontrivial dcpos necessarily lack decidable equality in our constructive setting. The fact that nontrivial dcpos are necessarily large has the important consequence that Tarski's theorem (and similar results) cannot be applied in nontrivial instances, even though it has a predicative proof. Further, we explained, by studying the large $\mathcal{V}$-sup-lattice of ordinals in a univalent universe $\mathcal{V}$, that generalisations of Tarski's theorem that allow for large structures are provably false. Finally, we elaborated on the connections between requiring suprema of families and of subsets in our predicative setting.

Moreover, we contributed to the overall theory of (predicative) univalent foundations by studying a set replacement principle and the connections and universe levels of set quotients and propositional truncations. We also presented a general criterion for decidable equality of indexed W-types that we applied to the syntax of PCF when proving that totality of PCF terms of base type is semidecidable.

8.2 Future work

To prove that D_∞ had a small compact basis, we used that each D_n is a $\mathcal{U}_0$-sup-lattice, so that we could apply the results of Section 4.13. Example 4.9.2 tells us that $\mathcal{L}_{\mathcal{U}_0}(\mathbb{N})$ has a small compact basis too, but to prove that the $\mathcal{U}_0$-dpcos in the Scott model of PCF have small compact bases using the techniques of Section 4.13, we would need $\mathcal{L}_{\mathcal{U}_0}(\mathbb{N})$ to be a $\mathcal{U}_0$-sup-lattice, but it isn't. However, it is complete for *bounded* families indexed by types in $\mathcal{U}_0$ and we believe that is possible to generalise the results of Section 4.13 from sup-lattices to bounded complete posets. Classically, this is fairly straightforward, but from preliminary considerations it appears that constructively one needs to impose certain decidability criteria on the bases of the dcpos. For instance that the partial order is decidable when restricted to basis elements. We also studied such decidability conditions in our paper [dJon21a] discussed below. These conditions should be satisfied by the bases of the dcpos in the Scott model of PCF, but we leave a full treatment of bounded complete dcpos with bases satisfying such conditions for future investigations.

It would be worthwhile to further develop domain-theoretic applications. Specifically, it would be interesting to give a fully rigorous formalisation of the surprising domain-theoretic algorithms that exhaustively search infinite sets in finite time due to Berger [Ber90] and Escardó [Esc08]. The fact that our development is constructive might then pay off as we could use our constructive proofs of domain theoretic facts to directly compute the output of such algorithms.

To complement the applications of domain theory in the semantics of programming language, it would be desirable to explore applications in (pointfree) topology. For example, can we predicatively replicate the connection [Hyl81] between exponentiable locales and continuous lattices?

Although some results (see Remark 3.3.3) can be stated in terms of opens of the Scott topology, we have not given a constructive account of the Scott topology. We did study this topic and a related apartness relation [BV11] in our paper [dJon21a]. To simplify the development and in the interest of appealing to a broader audience of constructivists, the work [dJon21a] is situated in informal constructive, but impredicative, set theory rather than univalent foundations. For this reason (and for time and space limitations) this paper is not part of this thesis, even though it shares the domain-theoretic theme.

Furthermore, we hope that our formalisation efforts (as discussed in Chapter 7) provide adequate support for those wishing to further develop computer-verified domain theory in univalent foundations. This hope is reinforced by the fact that a gifted MSci student, Brendan Hart, supervised by Martín Escardó and myself, was able to do so for a final year project [Har20] using an earlier and rudimentary version of our AGDA code.

Finally, the most fundamental and pressing question regarding predicativity in univalent foundations is whether propositional resizing can be given a computational interpretation.

Bibliography

[AAG04] Michael Abbott, Thorsten Altenkirch and Neil Ghani. 'Representing Nested Inductive Types Using W-Types'. In: *Automata, Languages and Programming: 31st International Colloquium, ICALP 2004, Turku, Finland, July 2004, Proceedings.* Ed. by Josep Díaz, Juhani Karhumäki, Arto Lepistö and Donald Sannella. Vol. 3142. Lecture Notes in Computer Science. Springer, 2004, pp. 59–71. DOI: 10.1007/978-3-540-27836-8_8 (cit. on p. 43).

[ACF+a] Carlo Angiuli, Evan Cavallo, Kuen-Bang Hou (Favonia), Robert Harper, Anders Mörtberg and Jon Sterling. *redtt*. https://redprl.org/#redtt. URL: https://github.com/redprl/redtt (cit. on p. 1).

[ACF+b] Carlo Angiuli, Evan Cavallo, Kuen-Bang Hou (Favonia), Robert Harper, Reed Mullanix and Jon Sterling. *cooltt*. https://redprl.org/#cooltt. URL: https://github.com/redprl/cooltt (cit. on p. 1).

[ACF+c] Carlo Angiuli, Evan Cavallo, Kuen-Bang Hou (Favonia), Robert Harper and Jon Sterling. *RedPRL*. https://redprl.org/#redprl. URL: https://github.com/redprl/sml-redprl (cit. on p. 1).

[Acz06] Peter Aczel. 'Aspects of general topology in constructive set theory'. In: *Annals of Pure and Applied Logic* 137.1–3 (2006): *Papers presented at the 2nd Workshop on Formal Topology (2WFTop 2002)*. Ed. by Bernhard Banaschewski, Thierry Coquand and Giovanni Sambin, pp. 3–29. DOI: 10.1016/j.apal.2005.05.016 (cit. on pp. 5, 112).

[Acz77] Peter Aczel. 'An Introduction to Inductive Definitions'. In: *Handbook of Mathematical Logic*. Ed. by Jon Barwise. Vol. 90. Studies in Logic and the Foundations of Mathematics. North-Holland, 1977, pp. 739–782. DOI: 10.1016/S0049-237X(08)71120-0.

[ADK17] Thorsten Altenkirch, Nils Anders Danielsson and Nicolai 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.*

Ed. by Javier Esparza and Andrzej S. Murawski. Vol. 10203. Lecture Notes in Computer Science. Springer, 2017, pp. 534–549. DOI: 10.1007/978-3-662-54458-7_31 (cit. on pp. 78, 126).

[AdMKU+] Jeremy Avigad, Leonardo de Moura, Soonho Kong, Sebastian Ullrich et al. *Theorem Proving in Lean 4*. URL: https://leanprover.github.io/theorem_proving_in_lean4/title_page.html (cit. on pp. 1, 4).

[AFS18] Steve Awodey, Jonas Frey and Sam Speight. 'Impredicative Encodings of (Higher) Inductive Types'. In: *LICS '18: Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*. Association for Computing Machinery, 2018, pp. 76–85. DOI: 10.1145/3209108.3209130 (cit. on p. 36).

[AGH+15] Thorsten Altenkirch, Neil Ghani, Peter Hancock, Conor McBride and Peter Morris. 'Indexed containers'. In: *Journal of Functional Programming* 25 (2015), e5. DOI: 10.1017/S095679681500009X (cit. on p. 43).

[AJ94] Samson Abramsky and Achim Jung. 'Domain theory'. In: *Handbook of Logic in Computer Science*. Ed. by S. Abramsky, Dov M. Gabray and T. S. E. Maibaum. Vol. 3. Clarendon Press, 1994, pp. 1–168. Updated online version available at https://www.cs.bham.ac.uk/~axj/pub/papers/handy1.pdf (cit. on pp. 1, 4, 8, 51, 68, 77, 79, 92, 94, 112).

[AKS15] Benedikt Ahrens, Krzysztof Kapulkin and Michael Shulman. 'Univalent categories and the Rezk completion'. In: *Mathematical Structures in Computer Science* 25.5 (2015): *From type theory and homotopy theory to Univalent Foundations of Mathematics*, pp. 1010–1039. DOI: 10.1017/S0960129514000486 (cit. on pp. 86, 89, 91).

[ANST20] Benedikt Ahrens, Paige Randall North, Michael Shulman and Dimitris Tsementzis. 'A Higher Structure Identity Principle'. In: *LICS '20: Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*. Association for Computing Machinery, 2020, pp. 53–66. DOI: 10.1145/3373718.3394755 (cit. on p. 2).

[AR10] Peter Aczel and Michael Rathjen. 'Notes on Constructive Set Theory'. https://www1.maths.leeds.ac.uk/~rathjen/book.pdf. Book draft. Aug. 2010 (cit. on p. 6).

[BBC+22] Marc Bezem, Ulrik Buchholtz, Pierre Cagne, Bjørn Ian Dundas and Daniel R. Grayson. 'Symmetry'. https://github.com/UniMath/SymmetryBook. Book draft. 31st Mar. 2022. Commit: 2722568 (cit. on pp. 1, 42).

[BCDE20] Marc Bezem, Thierry Coquand, Peter Dybjer and Martín Hötzel Escardó. *The Burali–Forti argument in HoTT/UF in Agda notation*. https://www.cs.bham.ac.uk/~mhe/TypeTopology/Published.Ordinals.BuraliForti.html. 2020.

https://github.com/martinescardo/TypeTopology/tree/master/source/Ordinals/BuraliForti.lagda, commit d59d10b (cit. on p. 141).

[Bee85] Michael J. Beeson. *Foundations of Constructive Mathematics: Metamathematical Studies.* Vol. 6. Ergebnisse der Mathematik und ihrer Grenzgebiete. 3. Folge / A Series of Modern Surveys in Mathematics. Springer-Verlag, 1985. DOI: 10.1007/978-3-642-68952-9 (cit. on p. 3).

[Bel97] J. L. Bell. 'Zorn's Lemma and Complete Boolean Algebras in Intuitionistic Type Theories'. In: *The Journal of Symbolic Logic* 62.4 (1997), pp. 1265–1279. DOI: 10.2307/2275642 (cit. on p. 140).

[Ber90] Ulrich Berger. 'Totale Objekte und Mengen in der Bereichstheorie'. PhD thesis. Department of Mathematics, Ludwig-Maximilians-Universität München (LMU Munich), 1990 (cit. on pp. 1, 152).

[BFC+] Guillaume Brunerie, Kuen-Bang Hou (Favonia), Evan Cavallo, Tim Baumann, Eric Finster, Jesper Cockx, Christian Sattler et al. *Homotopy Type Theory in Agda.* https://github.com/HoTT/HoTT-Agda (cit. on p. 1).

[BGL+17] Andrej Bauer, Jason Gross, Peter LeFanu Lumsdaine, Michael Shulman, Matthieu Sozeau and Bas Spitters. 'The HoTT Library: A Formalization of Homotopy Type Theory in Coq'. In: *CPP 2017: Proceedings of the 6th ACM SIGPLAN Conference on Certified Programs and Proofs.* Association for Computing Machinery, 2017, pp. 164–172. DOI: 10.1145/3018610.3018615 (cit. on p. 1).

[Bis67] Erret Bishop. *Foundations of Constructive Analysis.* McGraw-Hill Book Company, 1967 (cit. on pp. 2, 28).

[Bis70] Errett Bishop. 'Mathematics as a Numerical Language'. In: *Intuitionism and Proof Theory: Proceedings of the Summer Conference at Buffalo N. Y. 1968.* Ed. by A. Kino, J. Myhill and R. E. Vesley. Vol. 60. Studies in Logic and the Foundations of Mathematics. North-Holland Publishing Company, 1970, pp. 53–71. DOI: 10.1016/S0049-237X(08)70740-7 (cit. on p. 3).

[BK09] Andrej Bauer and Iztok Kavkler. 'A constructive theory of continuous domains suitable for implementation'. In: *Annals of Pure and Applied Logic* 159.3 (2009): *Joint Workshop Domains VIII — Computability over Continuous Data Types, Novosibirsk, September 11–15, 2007.* Ed. by Yuri L. Ershov, Klaus Keimel, Ulrich Kohlenbach and Andrei Morozov, pp. 251–267. DOI: 10.1016/j.apal.2008.09.025 (cit. on p. 5).

[BKV09] Nick Benton, Andrew Kennedy and Carsten Varming. 'Some Domain Theory and Denotational Semantics in Coq'. In: *Theorem Proving in Higher Order Logics: 22nd International Conference, TPHOLs 2009, Munich, Germany, August 17–20, 2009, Proceedings.* Ed. by Stefan Berghofer, Tobias Nipkow, Christian Urban and Makarius Wenzel. Vol. 5674. Lecture Notes in Computer Science. Springer, 2009, pp. 115–130. DOI: 10.1007/978-3-642-03359-9_10 (cit. on p. 5).

[BR87] Douglas Bridges and Fred Richman. *Varieties of Constructive Mathematics.* Vol. 97. London Mathematical Society Lecture Note Series. Cambridge University Press, 1987. DOI: 10.1017/CBO9780511565663 (cit. on pp. 28, 129).

[BV11] Douglas S. Bridges and Luminiţa Simona Vîţă. *Apartness and Uniformity: A Constructive Development.* Springer, 2011. DOI: 10.1007/978-3-642-22415-7 (cit. on p. 153).

[Cap05] Venanzio Capretta. 'General Recursion via Coinductive Types'. In: *Logical Methods in Computer Science* 1.2 (2005). DOI: 10.2168/LMCS-1(2:1)2005 (cit. on p. 78).

[CCHM18] Cyril Cohen, Thierry Coquand, Simon Huber and Anders Mörtberg. 'Cubical Type Theory: A Constructive Interpretation of the Univalence Axiom'. In: *21st International Conference on Types for Proofs and Programs (TYPES 2015).* Ed. by Tarmo Uustalu. Vol. 69. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2018, 5:1–5:34. DOI: 10.4230/LIPIcs.TYPES.2015.5 (cit. on pp. 2–3, 6).

[CD13] Thierry Coquand and Nils Anders Danielsson. 'Isomorphism is equality'. In: *Indagationes Mathematicae* 24.4 (2013): *In memory of N. G. (Dick) de Bruijn (1918–2012).* Ed. by J. W. Klop, R. Tijdeman and J. J. O. O. Wiegerinck. In memory of N.G. (Dick) de Bruijn (1918–2012), pp. 1105–1120. DOI: 10.1016/j.indag.2013.09.002 (cit. on p. 2).

[CH88] Thierry Coquand and Gérard Huet. 'The calculus of constructions'. In: *Information and Computation* 76.2–3 (1988), pp. 95–120. DOI: 10.1016/0890-5401(88)90005-3 (cit. on pp. 3, 10).

[CMR17] Thierry Coquand, Bassel Mannaa and Fabian Ruch. 'Stack semantics of type theory'. In: *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS).* IEEE, 2017, pp. 1–12. DOI: 10.1109/LICS.2017.8005130 (cit. on p. 78).

[Coq] The Coq Team. *The Coq Proof Assistant.* URL: https://coq.inria.fr (cit. on pp. 1, 4, 10–11, 147).

[Coq18] Thierry Coquand. 'A survey of constructive presheaf models of univalence'. In: *ACM SISLOG News* 5.3 (2018), pp. 54–65. DOI: 10.1145/3242953.3242962 (cit. on p. 78).

[CR12] Giovanni Curi and Michael Rathjen.
'Formal Baire Space in Constructive Set Theory'.
In: *Logic, Construction, Computation*. Ed. by Ulrich Berger,
Hannes Diener, Peter Schuster and Monika Seisenberger. Vol. 3.
Ontos Matematical Logic. De Gruyter, 2012, pp. 123–136.
DOI: 10.1515/9783110324921.123 (cit. on p. 6).

[CS18] Francesco Ciraulo and Giovanni Sambin.
'Embedding locales and formal topologies into positive topologies'.
In: *Archive for Mathematical Logic* 57 (2018), pp. 755–768.
DOI: 10.1007/s00153-017-0605-0 (cit. on p. 134).

[CSSV03] Thierry Coquand, Giovanni Sambin, Jan Smith and Silvio Valentini.
'Inductively generated formal topologies'.
In: *Annals of Pure and Applied Logic* 124.1–3 (2003), pp. 71–106.
DOI: 10.1016/s0168-0072(03)00052-6 (cit. on p. 4).

[Cur10a] Giovanni Curi.
'On some peculiar aspects of the constructive theory of point-free spaces'.
In: *Mathematical Logic Quarterly* 56.4 (2010), pp. 375–387.
DOI: 10.1002/malq.200910037 (cit. on pp. 6–7).

[Cur10b] Giovanni Curi. 'On the existence of Stone–Čech compactification'.
In: *The Journal of Symbolic Logic* 75.4 (2010), pp. 1137–1146.
DOI: 10.2178/jsl/1286198140 (cit. on pp. 6–7).

[Cur15] Giovanni Curi. 'On Tarski's fixed point theorem'. In: *Proceedings of the American Mathematical Society* 143.10 (2015), pp. 4439–4455.
DOI: 10.1090/proc/12569 (cit. on pp. 6–7).

[Cur18] Giovanni Curi. 'Abstract inductive and co-inductive definitions'.
In: *The Journal of Symbolic Logic* 83.2 (2018), pp. 598–616.
DOI: 10.1017/jsl.2018.13 (cit. on pp. 6–7).

[CUV19] James Chapman, Tarmo Uustalu and Niccolò Veltri.
'Quotienting the delay monad by weak bisimilarity'.
In: *Mathematical Structures in Computer Science* 29.1 (2019): *Best Papers Presented at ICTAC 2015*. Published online: 17th Oct. 2017, pp. 67–92.
DOI: 10.1017/S0960129517000184 (cit. on pp. 78, 126).

[CV16] Francesco Ciraulo and Steven Vickers. 'Positivity relations on a locale'.
In: *Annals of Pure and Applied Logic* 167.9 (2016): *Fourth Workshop on Formal Topology (4WFTop)*. Ed. by Thierry Coquand,
Maria Emilia Maietti, Giovanni Sambin and Peter Schuster, pp. 806–819.
DOI: 10.1016/j.apal.2016.04.009 (cit. on p. 134).

[dJE21a] Tom de Jong and Martín Hötzel Escardó.
'Domain Theory in Constructive and Predicative Univalent Foundations'.
In: *29th EACSL Annual Conference on Computer Science Logic (CSL 2021)*.
Ed. by Christel Baier and Jean Goubault-Larrecq. Vol. 183.
Leibniz International Proceedings in Informatics (LIPIcs).
Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2021, 28:1–28:18.

DOI: 10.4230/LIPIcs.CSL.2021.28. Expanded version with full proofs available on arXiv: 2008.01422 [math.LO] (cit. on pp. 8, 77, 111–112, 128, 140).

[dJE21b] Tom de Jong and Martín Hötzel Escardó. 'Predicative Aspects of Order Theory in Univalent Foundations'. In: *6th International Conference on Formal Structures for Computation and Deduction (FSCD 2021)*. Ed. by Naoki Kobayashi. Vol. 195. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2021, 8:1–8:18. DOI: 10.4230/LIPIcs.FSCD.2021.8 (cit. on pp. 7–9, 34, 50, 146, 159).

[dJE21c] Tom de Jong and Martín Hötzel Escardó. *Set quotients in univalent mathematics with a parameter for the universes of propositional truncations*. https://www.cs.bham.ac.uk/~mhe/TypeTopology/Published.UF.Quotient-F.html. AGDA development. 2021. https://github.com/martinescardo/TypeTopology/tree/master/source/UF/Quotient-F.lagda, commit d59d10b (cit. on p. 148).

[dJE22a] Tom de Jong and Martín Hötzel Escardó. 'On Small Types in Univalent Foundations'. Sept. 2022. arXiv: 2111.00482 [cs.LO]. Revised and expanded version of [dJE21b]. Accepted pending minor revision to a special issue of *Logical Methods in Computer Science* on selected papers from *FSCD 2021*. (Cit. on pp. 7–9, 34, 50, 139, 146).

[dJE22b] Tom de Jong and Martín Hötzel Escardó. *The ordinal of small ordinals has small suprema*. https://www.cs.bham.ac.uk/~mhe/TypeTopology/Published.Ordinals.OrdinalOfOrdinalsSuprema.html. AGDA development. 2022. https://github.com/martinescardo/TypeTopology/tree/master/source/Ordinals/OrdinalOfOrdinalsSuprema.lagda, commit d59d10b (cit. on p. 148).

[dJon19] Tom de Jong. *Coq/UniMath formalisation of the Scott model of PCF in univalent type theory*. https://tdejong.com/Scott-PCF-UniMath/UniMath.Partiality.PCF.html. 2019. DOI: 10.25500/edata.bham.00000912. URL: https://github.com/tomdjong/UniMath/tree/paper (cit. on pp. 1, 148–149).

[dJon21a] Tom de Jong. 'Sharp Elements and Apartness in Domains'. In: *37th Conference on the Mathematical Foundations of Programming Semantics (MFPS 2021)*. Ed. by Ana Sokolova. Vol. 351. Electronic Proceedings in Theoretical Computer Science. Open Publishing Association, 2021, pp. 134–151. DOI: 10.4204/EPTCS.351.9 (cit. on pp. 152–153).

[dJon21b] Tom de Jong. 'The Scott model of PCF in univalent type theory'. In: *Mathematical Structures in Computer Science* 31.10 (2021): *Homotopy Type Theory 2019*, pp. 1270–1300. DOI: 10.1017/S0960129521000153 (cit. on pp. 1, 7–8, 50, 77, 128).

[dJon22a] Tom de Jong. *Formalisation of domain theory in univalent mathematics in Agda.* https://www.cs.bham.ac.uk/~mhe/TypeTopology/Published.DomainTheory.index.html. 2022. DOI: 10.25500/edata.bham.00000912. https://github.com/martinescardo/TypeTopology/tree/master/source/DomainTheory, commit d59d10b (cit. on pp. 1, 147–150).

[dJon22b] Tom de Jong. *Propositional truncations and set replacement from small set quotients in univalent mathematics.* https://www.cs.bham.ac.uk/~mhe/TypeTopology/Published.UF.Quotient.html. Agda development. 2022. https://github.com/martinescardo/TypeTopology/tree/master/source/UF/Quotient.lagda, commit d59d10b (cit. on p. 148).

[dJon22c] Tom de Jong. 'Semidecidability: constructive taboos, choice principles and closure properties'. https://www.cs.bham.ac.uk/~mhe/TypeTopology/Published.NotionsOfDecidability.SemiDecidable.html. Agda development. 2022. https://github.com/martinescardo/TypeTopology/tree/master/source/NotionsOfDecidability/SemiDecidable.lagda, commit d59d10b (cit. on pp. 31, 126).

[dJon22d] Tom de Jong. *Small set quotients in univalent mathematics using set replacement.* https://www.cs.bham.ac.uk/~mhe/TypeTopology/Published.UF.Quotient-Replacement.html. Agda development. 2022. https://github.com/martinescardo/TypeTopology/tree/master/source/UF/Quotient-Replacement.lagda, commit d59d10b (cit. on p. 148).

[Doc14] Robert Dockins. 'Formalized, Effective Domain Theory in Coq'. In: *Interactive Theorem Proving: 5th International Conference, ITP 2014, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 14–17, 2014, Proceedings.* Ed. by Gerwin Klein and Ruben Gamboa. Vol. 8558. Lecture Notes in Computer Science. Springer, 2014, pp. 209–225. DOI: 10.1007/978-3-319-08970-6_14 (cit. on p. 5).

[EK17] Martín H. Escardó and Cory M. Knapp. 'Partial Elements and Recursion via Dominances in Univalent Type Theory'. In: *26th EACSL Annual Conference on Computer Science Logic (CSL 2017).* Ed. by Valentin Goranko and Mads Dam. Vol. 82. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2017, 21:1–21:16. DOI: 10.4230/LIPIcs.CSL.2017.21 (cit. on pp. 8, 31, 57, 59–62, 78, 126).

[Esc+] Martín Hötzel Escardó et al. *TypeTopology*.
https://www.cs.bham.ac.uk/~mhe/TypeTopology/index.html.
Agda development.
https://github.com/martinescardo/TypeTopology, commit d59d10b
(cit. on pp. 4, 46, 61, 96, 137, 141, 145, 147–148).

[Esc03] Martín H. Escardó. 'Joins in the Frame of Nuclei'.
In: *Applied Categorical Structures* 11 (2003), pp. 117–124.
DOI: 10.1023/A:1023555514029 (cit. on p. 139).

[Esc08] Martín Escardó. 'Exhaustible sets in higher-type computation'.
In: *Logical Methods in Computer Science* 4.3 (2008).
DOI: 10.2168/LMCS-4(3:3)2008 (cit. on pp. 1, 152).

[Esc18] Martín Hötzel Escardó.
Set quotients in univalent mathematics in Agda notation. https://www.cs.bham.ac.uk/~mhe/TypeTopology/Published.UF.Large-Quotient.html. 2018.
https://github.com/martinescardo/TypeTopology/tree/master/source/UF/Large-Quotient.lagda, commit d59d10b (cit. on p. 148).

[Esc19a] Martín Hötzel Escardó. 'Compact, totally separated and well-ordered types in univalent mathematics'.
https://www.ii.uib.no/~bezem/abstracts/TYPES_2019_paper_3.
Abstract for *TYPES*. June 2019.
HTML rendering of related Agda development available at
https://www.cs.bham.ac.uk/~mhe/TypeTopology/Types2019.html
(cit. on p. 46).

[Esc19b] Martín Hötzel Escardó.
'Introduction to Univalent Foundations of Mathematics with Agda'.
Nov. 2019. arXiv: 1911.00580 [cs.LO]
(cit. on pp. 1–2, 7, 11, 14, 17, 20–21, 29–31, 34–37, 40, 50, 61, 148).

[Esc21] Martín Hötzel Escardó. 'Injective types in univalent mathematics'.
In: *Mathematical Structures in Computer Science* 31.1 (2021), pp. 89–111.
DOI: 10.1017/s0960129520000225 (cit. on pp. 11, 50, 61).

[EX15] Martín Hötzel Escardó and Chuangjie 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)*. Ed. by Thorsten Altenkirch. Vol. 38.
Leibniz International Proceedings in Informatics (LIPIcs).
Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2015, pp. 153–164.
DOI: 10.4230/LIPIcs.TLCA.2015.153.

[Fre64] Peter Freyd. *Abelian Categories: An Introduction to the Theory of Functors*.
Harper & Row, 1964.
Republished in *Reprints in Theory and Applications of Categories*, Vol. 3, 2003. http://www.tac.mta.ca/tac/reprints/articles/3/tr3.pdf
(cit. on p. 130).

[GH04] Nicola Gambino and Martin Hyland. 'Wellfounded Trees and Dependent Polynomial Functors'. In: *Types for Proofs and Programs: International Workshop, TYPES 2003, Torino, Italy, April/May 2003, Revised Selected Papers.* Ed. by Stefano Berardi, Mario Coppo and Ferruccio Damiani. Vol. 3085. Lecture Notes in Computer Science. Springer, 2004, pp. 210–225. DOI: 10.1007/978-3-540-24849-1_14 (cit. on p. 43).

[GHK+03] G. Gierz, K. H. Hofmann, K. Keimel, J. D. Lawson, M. Mislove and D. S. Scott. *Continuous Lattices and Domains.* Vol. 93. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2003. DOI: 10.1017/CBO9780511542725 (cit. on pp. 1, 4–5, 51, 77, 79, 92, 94, 108).

[Gir71] Jean-Yves Girard. 'Une extension de l'interpretation fonctionnelle de Gödel à l'analyse et son application à l'élimination des coupures dans l'analyse et la théorie des types'. In: *Proceedings of the Second Scandinavian Logic Symposium.* Ed. by J. E. Fenstad. Vol. 63. Studies in Logic and the Foundations of Mathematics. North-Holland Publishing Company, 1971, pp. 63–92. DOI: 10.1016/S0049-237X(08)70843-7 (cit. on p. 36).

[GV72] A. Grothendieck and J. L. Verdier. 'Prefaisceaux'. In: *Théorie des Topos et Cohomologie Etale des Schémas.* Tome 1 (Séminaire de Géométrie Algébrique du Bois Marie 1963/1964, SGA4). Ed. by A. Dold. Heidelberg and B. Eckmann. Vol. 269. Lecture Notes in Mathematics. Springer, 1972, pp. 1–217. DOI: 10.1007/BFB0081552 (cit. on p. 84).

[Har20] Brendan Hart. 'Investigating Properties of PCF in Agda'. Final year MSci project. School of Computer Science, University of Birmingham, 2020. URL: https://raw.githubusercontent.com/BrendanHart/Investigating-Properties-of-PCF/master/InvestigatingPropertiesOfPCFInAgda.pdf. Agda code available at https://github.com/BrendanHart/Investigating-Properties-of-PCF (cit. on pp. 1, 77, 147–149, 153).

[Hed96] Michael Hedberg. 'A type-theoretic interpretation of constructive domain theory'. In: *Journal of Automated Reasoning* 16 (1996), pp. 369–425. DOI: 10.1007/BF00252182 (cit. on p. 5).

[Hed98] Michael Hedberg. 'A coherence theorem for Martin-Löf's type theory'. In: *Journal of Functional Programming* 8.4 (1998), pp. 413–436. DOI: 10.1017/S0956796898003153 (cit. on pp. 15–16, 28).

[HP08] J. Roger Hindley and Jonathan P. Seldin. *Lambda-Calculus and Combinators: an Introduction.* 2nd ed. Cambridge University Press, 2008. DOI: 10.1017/CBO9780511809835 (cit. on p. 119).

[Hug17a] Jasper Hugunin. *Characterizing the equality of Indexed W types.* Post on the *Homotopy Type Theory* mailing list. Sept. 2017. URL: https://groups.google.com/d/msg/homotopytypetheory/qj2OvRvqf-Q/hGFBczJGAwAJ (cit. on pp. 7, 47, 50).

[Hug17b] Jasper Hugunin. 'IWTypes — A Coq development of the theory of Indexed W types with function extensionality'. https://github.com/jashug/IWTypes. 2017 (cit. on pp. 7, 47, 50).

[Hyl81] J. M. E. Hyland. 'Function spaces in the category of locales'. In: *Continuous Lattices: Proceedings of the Conference on Topological and Categorical Aspects of Continuous Lattices (Workshop IV) Held at the University of Bremen, Germany, November 9–11, 1979.* Ed. by Bernhard Banaschewski and Rudolf-Eberhard Hoffmann. Vol. 871. Lecture Notes in Mathematics. Springer, 1981, pp. 264–281. DOI: 10.1007/BFB0089910 (cit. on p. 152).

[Hyl91] J. M. E. Hyland. 'First steps in synthetic domain theory'. In: *Category Theory: Proceedings of the International Conference held in Como, Italy, July 22–28, 1990.* Ed. by Aurelio Carboni, Maria Cristina Pedicchio and Guiseppe Rosolini. Vol. 1488. Lecture Notes in Mathematics. Springer, 1991, pp. 131–156. DOI: 10.1007/BFB0084217 (cit. on p. 5).

[Ish06] Hajime Ishihara. 'Reverse Mathematics in Bishop's Constructive Mathematics'. In: *Philosophia Scientiae* CS 6 (2006), pp. 43–59. DOI: 10.4000/philosophiascientiae.406 (cit. on pp. 7, 129).

[Jet] JetBrains. *Arend Theorem Prover.* https://arend-lang.github.io/. URL: https://github.com/JetBrains/Arend (cit. on p. 1).

[JJ82] Peter Johnstone and André Joyal. 'Continuous categories and exponentiable toposes'. In: *Journal of Pure and Applied Algebra* 25.3 (1982), pp. 255–296. DOI: 10.1016/0022-4049(82)90083-4 (cit. on pp. 5, 8, 79, 84, 86–87, 91, 112).

[Joh77] P. T. Johnstone. *Topos Theory.* Academic Press, 1977. Reprinted by Dover Publications in 2014 (cit. on p. 59).

[Joh84] Peter T. Johnstone. 'Open locales and exponentiation'. In: *Mathematical Applications of Category Theory.* Ed. by J. W. Gray. Vol. 30. Contemporary Mathematics. American Mathematical Society, 1984, pp. 84–116. DOI: 10.1090/conm/030/749770 (cit. on pp. 129, 134).

[Jun89] Achim Jung. *Cartesian Closed Categories of Domains.* Vol. 66. CWI Tracts. Centrum voor Wiskunde en Informatica (Centre for Mathematics and Computer Science), 1989. URL: https://ir.cwi.nl/pub/13191 (cit. on p. 108).

[Kaw17] Tatsuji Kawai. 'Geometric theories of patch and Lawson topologies'. 2017. arXiv: 1709.06403 [math.CT] (cit. on p. 4).

[Kaw21] Tatsuji Kawai. 'Predicative theories of continuous lattices'. In: *Logical Methods in Computer Science* 17.2 (2021). DOI: 10.23638/LMCS-17(2:22)2021 (cit. on p. 4).

[KECA17] Nicolai Kraus, Martín Hötzel Escardó, Thierry Coquand and Thorsten Altenkirch. '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 (cit. on pp. 2, 16, 23, 25, 36).

[KL21] Krzysztof Kapulkin and Peter LeFanu Lumsdaine. 'The simplicial model of Univalent Foundations (after Voevodsky)'. In: *Journal of the European Mathematical Society* 23.6 (2021), pp. 2071–2126. DOI: 10.4171/jems/1050 (cit. on pp. 6, 29, 31).

[Kna18] Cory Knapp. 'Partial Functions and Recursion in Univalent Type Theory'. PhD thesis. School of Computer Science, University of Birmingham, June 2018. URL: https://etheses.bham.ac.uk/id/eprint/8448/ (cit. on pp. 8, 31, 59, 78).

[KNX21] Nicolai Kraus, Fredrik Nordvall Forsberg and Chuangjie Xu. 'Connecting Constructive Notions of Ordinals in Homotopy Type Theory'. In: *46th International Symposium on Mathematical Foundations of Computer Science (MFCS 2021)*. Ed. by Filippo Bonchi and Simon J. Puglisi. Vol. 202. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2021, 70:1–70:16. DOI: 10.4230/LIPIcs.MFCS.2021.70 (cit. on p. 141).

[Koc91] Anders Kock. 'Algebras for the partial map classifier monad'. In: *Category Theory: Proceedings of the International Conference held in Como, Italy, July 22–28, 1990*. Ed. by Aurelio Carboni, Maria Cristina Pedicchio and Guiseppe Rosolini. Vol. 1488. Lecture Notes in Mathematics. Springer, 1991, pp. 262–278. DOI: 10.1007/BFB0084225 (cit. on p. 59).

[Kra15] Nicolai Kraus. 'The General Universal Property of the Propositional Truncation'. In: *20th International Conference on Types for Proofs and Programs (TYPES 2014)*. Ed. by Hugo Herbelin, Pierre Letouzey and Matthieu Sozeau. Vol. 39. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2015, pp. 111–145. DOI: 10.4230/LIPIcs.TYPES.2014.111 (cit. on p. 16).

[Lid20] David Lidell. 'Formalizing domain models of the typed and the untyped lambda calculus in Agda'. Master's thesis. Department of Computer Science and Engineering, Chalmers University of Technology and University of Gothenburg, 2020. URL: http://hdl.handle.net/2077/67193 (cit. on p. 5).

[LN15] John Longley and Dag Normann. *Higher-Order Computability*. Theory and Applications of Computability. Springer, 2015. DOI: 10.1007/978-3-662-47992-6 (cit. on pp. 1, 51).

[Lon95] John Longley. 'Realizability Toposes and Language Semantics'. PhD thesis. Department of Computer Science, University of Edinburgh, 1995. URL: https://www.lfcs.inf.ed.ac.uk/reports/95/ECS-LFCS-95-332 (cit. on p. 57).

[Mag94] Lena Magnusson. 'The Implementation of ALF: a Proof Editor Based on Martin-Löf's Monomorphic Type Theory with Explicit Substitution'. PhD thesis. Department of Computing Science, Chalmers University of Technology and University of Gothenburg, 1994. URL: http://hdl.handle.net/2077/12916 (cit. on p. 5).

[Mar75] Per Martin-Löf. 'An Intuitionistic Theory of Types: Predicative Part'. In: *Logic Colloquium '73: Proceedings of the Logic Colloquium, Bristol, July 1973*. Ed. by H. E. Rose and J. C. Shepherdson. Vol. 80. Studies in Logic and the Foundations of Mathematics. North-Holland Publishing Company and American Elsevier Publishing Company, 1975, pp. 73–118. DOI: 10.1016/S0049-237X(08)71945-1 (cit. on pp. 2, 10).

[Mar82] Per Martin-Löf. 'Constructive Mathematics and Computer Programming'. In: *Logic, Methodology and Philosophy of Science VI: Proceedings of the Sixth International Congress of Logic, Methodology and Philosophy of Science, Hannover, 1979*. Ed. by L. Jonathan Cohen, Jerzy Łoś, Helmut Pfeiffer and Klaus-Peter Podewski. Vol. 104. Studies in Logic and the Foundations of Mathematics. North-Holland Publishing Company and PWN — Polish Scientific Publishers, 1982, pp. 153–175. DOI: 10.1016/S0049-237X(09)70189-2 (cit. on p. 3).

[Mar84] Per Martin-Löf. *Intuitionistic Type Theory*. Vol. 1. Studies in Proof Theory. Notes by Giovanni Sambin of a series of lectures given in Padua, June 1980. Bibliopolis, 1984. Re-typeset and searchable version available at https://raw.githubusercontent.com/michaelt/martin-lof/master/pdfs/Bibliopolis-Book-retypeset-1984.pdf (cit. on pp. 1, 10, 43, 50).

[MP89] Michael Makkai and Robert Paré. *Accessible Categories: The Foundations of Categorical Model Theory*. Vol. 104. Contemporary Mathematics. American Mathematical Society, 1989. DOI: 10.1090/conm/104 (cit. on p. 112).

[MV04] Maria Emilia Maietti and Silvio Valentini. 'Exponentiation of Scott Formal Topologies'. In: *Electronic Notes in Theoretical Computer Science* 73 (2004): *Proceedings of the Workshop on Domains VI*. Ed. by Martín Escardó and Achim Jung, pp. 111–131. DOI: 10.1016/j.entcs.2004.08.005 (cit. on p. 4).

[NDCA+] Ulf Norell, Nils Anders Danielsson, Jesper Cockx, Andreas Abel et al. *Agda*. URL: https://wiki.portal.chalmers.se/agda/pmwiki.php (cit. on pp. 1, 4, 11, 147).

[Neg02] Sara Negri. 'Continuous domains as formal spaces'. In: *Mathematical Structures in Computer Science* 12.1 (2002), pp. 19–52. DOI: 10.1017/S0960129501003450 (cit. on p. 4).

[Pat97] Dito Pataraia. 'A constructive proof of Tarski's fixed-point theorem for dcpos'. Presented at the *65th Peripatetic Seminar on Sheaves and Logic*, Aarhus, Denmark. Nov. 1997 (cit. on p. 139).

[Pha91] Wesley Phao. 'Domain Theory in Realizability Toposes'. PhD thesis. Department of Computer Science, University of Edinburgh, 1991. URL: https://www.lfcs.inf.ed.ac.uk/reports/91/ECS-LFCS-91-171 (cit. on p. 57).

[Plo77] G.D. Plotkin. 'LCF considered as a programming language'. In: *Theoretical Computer Science* 5.3 (1977), pp. 223–255. DOI: 10.1016/0304-3975(77)90044-5 (cit. on pp. 1, 8, 51, 113, 118–119, 128).

[Plo83] Gordon Plotkin. *Domains*. Course notes known as the *Pisa notes*. 1983. URL: https://homepages.inf.ed.ac.uk/gdp/publications/Domains_a4.ps (cit. on p. 128).

[PM21] Dirk Pattinson and Mina Mohammadian. 'Constructive Domains with Classical Witnesses'. In: *Logical Methods in Computer Science* 17.1 (2021). DOI: 10.23638/LMCS-17(1:19)2021 (cit. on p. 5).

[PVW15] Álvaro Pelayo, Vladimir Voevodsky and Michael A. Warren. 'A univalent formalization of the p-adic numbers'. In: *Mathematical Structures in Computer Science* 25.5 (2015): *From type theory and homotopy theory to Univalent Foundations of Mathematics*, pp. 1147–1171. DOI: 10.1017/S0960129514000541 (cit. on p. 35).

[RBP+] Egbert Rijke, Elisabeth Bonnevier, Jonathan Prieto-Cubides et al. *Univalent mathematics in Agda*. https://unimath.github.io/agda-unimath/. URL: https://github.com/UniMath/agda-unimath/ (cit. on p. 1).

[Reu99] Bernhard Reus. 'Formalizing Synthetic Domain Theory'. In: *Journal of Automated Reasoning* 23 (1999), pp. 411–444. DOI: 10.1023/A:1006258506401 (cit. on p. 5).

[Rey74] John C. Reynolds. 'Towards a theory of type structure'. In: *Programming Symposium: Proceedings, Colloque sur la Programmation, Paris, April 9–11, 1974*. Ed. by G. Goos and J. Hartmanis. Vol. 19. Lecture Notes in Computer Science. Springer-Verlag, 1974, pp. 408–425. DOI: 10.1007/3-540-06859-7_148 (cit. on p. 36).

[Rij17] Egbert Rijke. 'The join construction'. Jan. 2017. arXiv: 1701.07538 [math.CT] (cit. on pp. 32, 42–43, 50).

[Rij18] Egbert Rijke. 'Classifying Types: Topics in synthetic homotopy theory'. PhD thesis. Department of Philosophy, Carnegie Mellon University, July 2018. arXiv: 1906.09435 [math.LO] (cit. on p. 1).

[Ros86] G. Rosolini. 'Continuity and effectiveness in topoi'. PhD thesis. University of Oxford, 1986. URL: https://www.researchgate.net/publication/35103849_Continuity_and_effectiveness_in_topoi (cit. on pp. 5, 57, 59).

[Ros87] G. Rosolini. 'Categories and effective computations'. In: *Category Theory and Computer Science: Edinburgh, U.K., September 7–9, 1987. Proceedings.* Ed. by David H. Pitt, Axel Poigné and David E. Rydeheard. Vol. 283. Lecture Notes in Computer Science. Springer, 1987. DOI: 10.1007/3-540-18508-9_17 (cit. on pp. 5, 59).

[RS15] Egbert Rijke and Bas Spitters. 'Sets in Homotopy Type Theory'. In: *Mathematical Structures in Computer Science* 25.5 (2015): *From type theory and homotopy theory to Univalent Foundations of Mathematics*, pp. 1172–1202. DOI: 10.1017/S0960129514000553 (cit. on p. 37).

[RS99] Bernhard Reus and Thomas Streicher. 'General synthetic domain theory — a logical approach'. In: *Mathematical Structures in Computer Science* 9.2 (1999), pp. 177–223. DOI: 10.1017/S096012959900273X (cit. on pp. 5, 59).

[Sam03] Giovanni Sambin. 'Some points in formal topology'. In: *Theoretical Computer Science* 305.1–3 (2003): *Topology in Computer Science*, pp. 347–408. DOI: 10.1016/S0304-3975(02)00704-1 (cit. on pp. 4, 134).

[Sam87] Giovanni Sambin. 'Intuitionistic formal spaces — a first communication'. In: *Mathematical logic and its applications.* Ed. by Dimiter G. Skordev. Springer, 1987, pp. 187–204. DOI: 10.1007/978-1-4613-0897-3_12 (cit. on p. 4).

[Sat15] Christian Sattler. 'On relating Indexed W-Types with Ordinary Ones'. https://cs.ioc.ee/types15/abstracts-book/contrib31.pdf. Abstract for *TYPES*. May 2015. Slides available at https://cs.ioc.ee/types15/slides/sattler-slides.pdf (cit. on p. 43).

[Sco70] Dana Scott. *Outline of a Mathematical Theory of Computation.* Tech. rep. PRG02. Oxford University Computing Laboratory, Nov. 1970. URL: https://www.cs.ox.ac.uk/publications/publication3720-abstract.html (cit. on pp. 3, 53, 80).

[Sco72] Dana S. Scott. 'Continuous lattices'. In: *Toposes, Algebraic Geometry and Logic: Dalhousie University, Halifax, January 16–19, 1971.* Ed. by F. W. Lawvere. Vol. 274. Lecture Notes in Mathematics. Springer, 1972, pp. 97–136. DOI: 10.1007/BFB0073967 (cit. on pp. 8, 51, 69, 71, 77, 113, 128).

[Sco82a] Dana S. Scott. 'Domains for denotational semantics'. In: *Automata, Languages and Programming: Ninth Colloquium. Aarhus, Denmark, July 1982.* Ed. by Mogens Nielsen and Erik Meineche Schmidt. Vol. 140. Lecture Notes in Computer Science. Springer-Verlag, 1982, pp. 577–610. DOI: 10.1007/BFB0012801 (cit. on p. 4).

[Sco82b] Dana S. Scott. 'Lectures on a Mathematical Theory of Computation'. In: *Theoretical Foundations of Programming Methodology: Lecture Notes of an International Summer School, directed by F. L. Bauer, E. W. Dijkstra and C. A. R. Hoare.* Ed. by Manfred Broy and Gunther Schmidt. Vol. 91. NATO Advanced Study Institutes Series. Springer, 1982, pp. 145–292. DOI: 10.1007/978-94-009-7893-5_9 (cit. on pp. 1, 4).

[Sco93] Dana S. Scott. 'A type-theoretical alternative to ISWIM, CUCH, OWHY'. In: *Theoretical Computer Science* 121.1–2 (1993), pp. 411–440. DOI: 10.1016/0304-3975(93)90095-B (cit. on pp. 1, 8, 51, 113, 118–119, 128).

[Shu11] Michael Shulman. *Topos Theory Can Make You a Predicativist.* Blog post. Jan. 2011. URL: https://golem.ph.utexas.edu/category/2011/01/topos_theory_can_make_you_a_pr.html (cit. on p. 3).

[Shu15] Michael Shulman. 'Univalence for inverse diagrams and homotopy canonicity'. In: *Mathematical Structures in Computer Science* 25.5 (2015): *From type theory and homotopy theory to Univalent Foundations of Mathematics*, pp. 1203–1277. DOI: 10.1017/S0960129514000565 (cit. on p. 6).

[Shu16] Michael Shulman. 'Idempotents in intensional type theory'. In: *Logical Methods in Computer Science* 12.3 (2016), pp. 1–24. DOI: 10.2168/LMCS-12(3:9)2016 (cit. on pp. 22, 34).

[Shu19] Michael Shulman. 'All $(\infty, 1)$-toposes have strict univalent universes'. Apr. 2019. arXiv: 1904.07004 [math.CT] (cit. on pp. 1–3, 6).

[Smy77] M. B. Smyth. 'Effectively given domains'. In: *Theoretical Computer Science* 5.3 (1977), pp. 257–274. DOI: 10.1016/0304-3975(77)90045-7 (cit. on pp. 3, 111–112).

[SP82] M. B. Smyth and G. D. Plotkin. 'The Category-Theoretic Solution of Recursive Domain Equations'. In: *SIAM Journal on Computing* 11.4 (1982), pp. 761–783. DOI: 10.1137/0211062.

[Str06] Thomas Streicher. *Domain-Theoretic Foundations of Functional Programming.* World Scientific, 2006. DOI: 10.1142/6284 (cit. on pp. 123, 128).

[Str93] Thomas Streicher. 'Investigations into intensional type theory'. Habilitationsschrift. Faculty of Mathematics, Ludwig-Maximilians-Universität München (LMU Munich), 1993. URL: https://www2.mathematik.tu-darmstadt.de/~streicher/HabilStreicher.pdf (cit. on p. 13).

[SVV96] Giovanni Sambin, Silvio Valentini and Paolo Virgili. 'Constructive domain theory as a branch of intuitionistic pointfree topology'. In: *Theoretical Computer Science* 159.2 (1996), pp. 319–341. DOI: 10.1016/0304-3975(95)00169-7 (cit. on p. 4).

[Swa19a] Andrew W Swan. 'Choice, Collection and Covering in Cubical Sets'. https://www.youtube.com/watch?v=r9KbEOzyr1g. Talk at *Homotopy Type Theory Electronic Seminar Talks (HoTTEST)*, online. 6th Nov. 2019. URL: https://www.math.uwo.ca/faculty/kapulkin/seminars/hottestfiles/Swan-2019-11-06-HoTTEST.pdf (cit. on pp. 6, 78).

[Swa19b] Andrew W Swan. 'Counterexamples in Cubical Sets'. Talk at *Mathematical Logic and Constructivity: The Scope and Limits of Neutral Constructivism*, Stockholm, Sweden. 20th Aug. 2019. URL: https://logic.math.su.se/mloc-2019/slides/Swan-mloc-2019-slides.pdf (cit. on pp. 6, 78).

[Tai67] W. W. Tait. 'Intensional Interpretations of Functionals of Finite Type I'. In: *The Journal of Symbolic Logic* 32.2 (1967), pp. 198–212. DOI: 10.2307/2271658 (cit. on p. 123).

[Tar55] Alfred Tarski. 'A lattice-theoretical fixpoint theorem and its applications'. In: *Pacific Journal of Mathematics* 5.2 (1955), pp. 285–309. DOI: 10.2140/pjm.1955.5.285 (cit. on p. 139).

[Tay99] Paul Taylor. *Practical Foundations of Mathematics*. Vol. 59. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 1999 (cit. on pp. 4, 53).

[TvD88] A. S. Troelstra and D. van Dalen. *Constructivism in Mathematics: An Introduction*. Volume II. Ed. by J. Barwise, D. Kaplan, H. J. Keisler, P. Suppes and A. S. Troelstra. Vol. 123. Studies in Logic and the Foundations of Mathematics. Elsevier, 1988 (cit. on p. 56).

[Uem19] Taichi Uemura. 'Cubical Assemblies, a Univalent and Impredicative Universe and a Failure of Propositional Resizing'. In: *24th International Conference on Types for Proofs and Programs (TYPES 2018)*. Ed. by Peter Dybjer, José Espírito Santo and Luís Pinto. Vol. 130. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2019, 7:1–7:20. DOI: 10.4230/LIPIcs.TYPES.2018.7 (cit. on pp. 6–7).

[Uni13] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study: https://homotopytypetheory.org/book, 2013 (cit. on pp. 1–2, 7, 11, 15–16, 20–21, 23, 31, 33, 37, 50, 127, 141–142).

[VAG+] Vladimir Voevodsky, Benedikt Ahrens, Daniel Grayson et al. *UniMath — a computer-checked library of univalent mathematics*. https://unimath.org. URL: https://github.com/UniMath/UniMath (cit. on p. 147).

[vdBer06] Benno van den Berg. 'Predicative topos theory and models for constructive set theory'. PhD thesis. Utrecht University, 2006. URL: https://staff.fnwi.uva.nl/b.vandenberg3/thesis/mythesis.pdf (cit. on p. 6).

[vdBer11] Benno van den Berg. 'Categorical semantics of constructive set theory'. Habilitationsschrift. Department of Mathematics, Technical University of Darmstadt, 2011. URL: https://staff.fnwi.uva.nl/b.vandenberg3/Habil/habil.pdf.

[VMA19] Andrea Vezzosi, Anders Mörtberg and Andreas Abel. 'Cubical Agda: A Dependently Typed Programming Language with Univalence and Higher Inductive Types'. In: *Proceedings of the ACM on Programming Languages* 3.ICFP, 87 (2019), pp. 1–29. DOI: 10.1145/3341691 (cit. on pp. 1–3).

[Voe11] Vladimir Voevodsky. 'Resizing Rules — their use and semantic justification'. Talk at *18th International Conference on Types for Proofs and Programs (TYPES)*, Bergen, Norway. 11th Sept. 2011. URL: https://www.math.ias.edu/vladimir/sites/math.ias.edu.vladimir/files/2011_Bergen.pdf (cit. on pp. 3, 6, 35).

[Voe15] Vladimir Voevodsky. 'An experimental library of formalized mathematics based on the univalent foundations'. In: *Mathematical Structures in Computer Science* 25.5 (2015): *From type theory and homotopy theory to Univalent Foundations of Mathematics*, pp. 1278–1294. DOI: 10.1017/S0960129514000577 (cit. on pp. 1–3, 147).

[vOos08] Jaap van Oosten. *Realizability: An Introduction to its Categorical Side*. Ed. by S. Abramsky, S. Artemov, D. M. Gabbay, A. Kechris, A. Pillay and R. A. Shore. Vol. 152. Studies in Logic and the Foundations of Mathematics. Elsevier, 2008. DOI: 10.1016/s0049-237X(08)X8001-2 (cit. on p. 57).

Index of symbols

Index