Thesis for the Degree of Licentiate of Philosophy

A Model of Type Theory in Cubical Sets

Simon Huber

UNIVERSITY OF GOTHENBURG

Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
Göteborg, Sweden 2015

A Model of Type Theory in Cubical Sets

SIMON HUBER

Technical Report 132L
ISSN 1652-876X
Department of Computer Science and Engineering
Programming Logic Research Group

Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-405 30 Göteborg
Sweden
Telephone +46 (0)31 786 0000

Printed at Chalmers Reproservice
Göteborg, Sweden 2015

Abstract

The intensional identity type is one if the most intricate concepts of dependent type theory. The recently discovered connection between homotopy theory and type theory gives a novel perspective on the identity type. Voevodsky's so-called Univalence Axiom furthermore explains the identity type for type theoretic universes as homotopy equivalences. This licentiate thesis is concerned with understanding these new developments from a computational point of view. While the Univalence Axiom has a model using Kan simplicial sets, this model inherently uses classical logic and thus can not be used to explain the axiom computationally. To preserve the computational properties of type theory it is, however, crucial to give a computational interpretation of the added constants. This thesis presents a model of dependent type theory with dependent products, sums, a universe, and identity types, based on *cubical sets*. The novelty of this model is that it is formulated in a constructive meta theory.

Acknowledgments

First and foremost I want to thank my supervisor Thierry Coquand for his continuous guidance and great support, and for the countless discussions at the white board or wherever else which taught me so much. I also would like to thank Nils Anders Danielsson for helping me out, especially during the time Thierry was in Princeton.

Thanks to my office mates Bassel Mannaa, Guilhem Moulin, and Anders Mörtberg for all the discussions and for being great friends! Thanks to Cyril Cohen, Thierry Coquand, and Anders Mörtberg for the intense and entertaining hacking sessions on Cubical. Thanks to the members of the Programming Logic group for the great scientific environment. Big thanks also to all the other members of the department and its students for making it such a nice place to work.

My gratitude to Marcelo Fiore for accepting to be my discussion leader, and to the members of my follow-up group. I would also like to thank the members of the "cubical seminar" at the Institut Henri Poincaré for letting me present my work and the valuable feedback. For discussions on cubes, parametricity, and what not, I would like to thank Thorsten Altenkirch, Marc Bezem, Jean-Philippe Bernardy, Guillaume Brunerie, Favonia, Ambrus Kaposi, Dan Licata, Peter Lumsdaine, Andrew Polonsky, Peter Schuster, Bas Spitters, Andrea Vezzosi, and everyone else I forgot to mention!

For their constant support and help I would like to thank my family and friends. In particular I would like to thank my nephews Lionel and Severin who always succeeded cheering me up! Many thanks also to Sofie Kindahl for always being there for me!

Contents

Introduction

Dependent type theory has been successfully used as a foundation to develop formalized mathematics and computer science. This is reflected not only by the popularity of the proof assistants Coq and Agda (among others) based on type theory, but also the formal proofs of the Four Color Theorem [20] and the Feit-Thompson (Odd Order) Theorem [21] show that non-trivial mathematics can be encoded in type theory.

There is however one especially intricate concept in dependent type theory: *equality*. The *identity type*, $\mathtt{Id}_A(u, v)$, for u and v of type A, serves as the type of proofs witnessing that u and v are *identical*. Dependent type theory also comes with a different notion of equality namely *definitional equality* which, in contrast to the identity type, is not a type (hence the identity type is often referred to as *propositional equality*), but definitional equality concerns the computational aspect of type theory, and is often presented as a judgment (and hence also referred to as *judgmental equality*). Martin-Löf formulated two variations of the identity type, the intensional identity type from [31], and the "extensional" identity type from [32]. In the latter one has the so-called *reflection rule*: if we have a proof of $\mathtt{Id}_A(u, v)$, we can deduce that u and v are definitionally equal. This equality reflection rule, however, destroys the decidability of type checking, resulting in a formal system where it is not decidable whether a given syntactic entity is in fact a proof of a given proposition. The intensional identity type on the other hand, drops this rule while still keeping its introduction and elimination rules (the latter often referred to as the J-rule), which entail the usual properties of equality, in particular Leibniz's rule of indiscernability of identicals.

While the formulation without equality reflection retains decidability of type checking, the intensional identity type per se is, however, often too restrictive to directly encode certain mathematical practices such as identifying two functions which are extensionally equal, that is, for which each argument gives equal results. The intensional identity type only identifies functions which are definitionally equal which often tends to be too restrictive in practice. One remedy to this particular problem is to simply add functional extensionality as an axiom—but this destroys the good computational properties of type theory: e.g., one can define a closed term of type N (the type of natural numbers) which is *not* definitionally equal to a numeral (i.e., a term merely built up from successors and zero). One possibility is to exploit Hofmann's [24] *setoid inter-*

pretation and work with *setoids* (a type with an equivalence relation) instead, and relativize to functions preserving this relation. This, however, turns out to be rather cumbersome in practice. A type theory incorporating ideas from the setoid interpretation and reconciling some of the extensional concepts with intensional type theory is observational type theory [2].

The rich structure of the intensional identity type stems from the fact that in type theory we can *iterate* the identity type to obtain "higher-dimensional" identity types: for p and q of type $\mathtt{Id}_A(u, v)$ we can form the $\mathtt{Id}_{\mathtt{Id}_A(u,v)}(p, q)$; and for α and β of type $\mathtt{Id}_{\mathtt{Id}_A(u,v)}(p, q)$ we can furthermore form the type

$$\mathtt{Id}_{\mathtt{Id}_{\mathtt{Id}_A(u,v)}(p,q)}(\alpha, \beta);$$

etc. It is a natural question to ask what the structure of these higher identity types is. In particular, can one prove $\mathtt{Id}_{\mathtt{Id}_A(u,v)}(p, q)$ for p and q of type $\mathtt{Id}_A(u, v)$, i.e., is there essentially only "one way" to prove a propositional equality $\mathtt{Id}_A(u, v)$? The latter principle is often referred to as *uniqueness of identity proofs* (UIP). In extensional type theory UIP is provable from the reflection rule and thus this hierarchy collapses. For intensional type theory, however, UIP is not provable as shown by the pioneering work of Hofmann and Streicher [27]. They devise a model of intensional type theory where a (closed) type A is interpreted by a *groupoid*[1] G and the closed terms of A are interpreted as the objects in G. The arrows $u \to v$ for u and v objects of G stand for the witnesses that u and v are propositionally equal. UIP is then refuted by specifying a groupoid with two distinct parallel arrows $u \to v$. The fact that G is required to be a groupoid stems from the fact that one can define–internally in type theory–operations corresponding to the groupoid operations: the introduction rule $\mathtt{refl}\, u : \mathtt{Id}_A(u, u)$ corresponds to the identity map, transitivity

$$\_ \circ \_ : \mathtt{Id}_A(v, w) \to \mathtt{Id}_A(u, v) \to \mathtt{Id}_A(u, w)$$

corresponds to composition, and symmetry

$$\_^{-1} : \mathtt{Id}_A(u, v) \to \mathtt{Id}_A(v, u)$$

corresponds to taking the inverse. These operations satisfy the expected groupoid equations, but in general only *up to propositional equality*; e.g., $(p \circ q) \circ r$ (for appropriately typed p, q, r) is not definitional equal to $p \circ (q \circ r)$, but we can define a *higher* equality between equalities

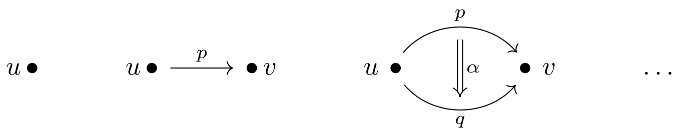
$$\alpha_{p,q,r} : \mathtt{Id}_{\mathtt{Id}_A(u,w')}((p \circ q) \circ r, p \circ (q \circ r)).$$

Already the groupoid interpretation suggests that a type in intensional type theory should be thought of more than merely a "set". Instead, a type should be thought of as a topological space—but up to *homotopy*. Around 2006 Awodey and Warren [3] and Garner [18] discovered connections between

[1] A category where each arrow is invertible.

homotopy theory and type theory in the context of Quillen model categories. Moreover, Streicher and independently Voevodsky [29] built a model of type theory using Kan simplicial sets (the latter model also supporting type theoretic universes, i.e., types of types).

One crucial point in this analogy between types and spaces is what under this view is the interpretation of the identity type: while $u : A$ should be thought of as a point in the space A, a term $p : \mathtt{Id}_A(u, v)$ should be thought of as a *path* between u and v, i.e., a continuous map $p\colon \mathbb{I} \to A$ (where $\mathbb{I} = [0, 1]$ is the interval) such that $p(0) = u$ and $p(1) = v$. Higher equalities $\alpha : \mathtt{Id}_{\mathtt{Id}_A(u,v)}(p, q)$ correspond to homotopies (or 2-dimensional paths) $\alpha\colon \mathbb{I} \to A^{\mathbb{I}} \cong \mathbb{I} \times \mathbb{I} \to A$ between the paths $p, q\colon \mathbb{I} \to A$ with $\alpha|_{\{0\}\times\mathbb{I}} = p$ and $\alpha|_{\{1\}\times\mathbb{I}} = q$; etc.



All the higher homotopies on a space A organize into a structure called the *fundamental* ∞*-groupoid* of A, the prime example of a so-called *weak* ∞*-groupoid*. These higher groupoids are closely connected to homotopy theory, as proposed by Grothendieck [23].

Another groundbreaking insight by Voevodsky was that the interpretation of type theory using Kan simplicial sets satisfies an additional axiom, the so-called *Univalence Axiom*. This axiom explains the identity type for type theoretic universes. (A type theoretic universe U is a type of types.) The Univalence Axiom states, loosely speaking, that isomorphic types (as elements of U) are equal, where "equal" refers to the identity type of U. This formalizes the everyday habit by mathematicians to identify isomorphic structures. Although this is clearly not valid in set theory: $0 \in \{0\}$ but $1 \notin \{0\}$ even though $\{0\} \cong \{1\}$. The Univalence Axiom can also be seen as an extensionality principle: it is a natural generalization of the concept in simple type theory formulated by Church [10] that two propositions are equal given that they are logically equivalent. The Univalence Axiom allows for a better encoding of mathematics in dependent type theory à la Martin-Löf, as is particularly shown in the book [42]. One aspect of this is that it entails the function extensionality axiom discussed above.

Another important contribution by Voevodsky is how notions from homotopy theory translate into the language of type theory. For example, the important hierarchy of *h-levels*: Call a type A *contractible*, or of h-level 0, if we have $(\Sigma a : A)(\Pi x : A)\ \mathtt{Id}_A(a, x)$. A type A is of h-level $n + 1$ if for all $x, y : A$ the type $\mathtt{Id}_A(x, y)$ is of h-level n. For a type to be of h-level 2 is then the same to have UIP for this type (those are called *h-sets*). Types with h-level 3 are called *h-groupoids*. The Univalence Axiom yields that this hierarchy is "well behaved", in particular it is closed under function types.

Similar to the situation with the function extensionality axiom, simply postulating the Univalence Axiom as a constant added to Martin-Löf type theory destroys the good computational behavior of type theory, making it necessary to explain univalence computationally. One possible attempt to do so is to build a model of this axiom in type theory itself or at least in a constructive metatheory. Such a computational interpretation could then be obtained through semantics, for example, by evaluating a term of type N (the natural numbers) in the model.

The model of univalence using Kan simplicial sets by Voevodsky is, as it is, not suited to justify the axiom computationally since it is *not* constructive. This model is formulated using ZFC set theory and uses classical logic and the axiom of choice in an essential way. One problem with using simplicial sets has to do with the fact that the notion of degeneracy is *not decidable* in general and that simplicial maps have to commute with degeneracy maps. The theory of simplicial sets and Kan simplicial sets however uses this fact crucially. One example where this is needed is that for a Kan fibration, a path in the base induces an equivalence between the fibers over the endpoints; this was shown in [6], using a Kripke counter-model, to be not intuitionistically provable. Similar problems seem to appear when looking at the different proofs (e.g., [33, 19]) of the fact that B^A is a Kan simplicial set if B is so.

One possible remedy for this problem is to use Kan *semi*-simplicial sets instead as was done in [4]. This approach however is very involved and does not model various laws for substitutions. This licentiate thesis presents a different approach based on *cubical sets*. Cubical sets were used to give the first *combinatorial* definition of homotopy groups by Kan [28].

Our formulation of cubical sets gives a *formal* representation of cubes seen as continuous maps $u\colon \mathbb{I}^J \to X$ (with $\mathbb{I} = [0,1]$) where J is a finite set of names $x_1, \dots, x_n$, instead of the more common $u\colon \mathbb{I}^n \to X$. We want to view such a cube as "value depending on the names" $x_1, \dots, x_n$. We have face operations like, e.g., $u(x_i = 0)$, setting the x_i-coordinate to 0; for a fresh name y we can view u as depending on $x_1, \dots, x_n, y$; this corresponds to the degeneracy operation; another primitive operation is to rename a variable. Thus the basic operations in cubical sets are certain substitutions of names. This formulation bears close resemblance to the theory of nominal sets [37, 36, 38]. There are various different variations of cubical sets used in the literature (see, e.g., [22]).

Formally cubical sets are given as presheaves on the (opposite of the) so-called cubical category, a category given by finite sets of names and certain substitutions between them. Following, e.g., [25], this yields a model of type theory where the contexts are interpreted as cubical sets. However to obtain the envisaged identity types we have to strengthen, similarly as in the Kan simplicial set model, the interpretation of types: we require types to have a *Kan structure*. This structure is a refinement of Kan's original extension condition ("Kan cubical sets") as defined in [28] which amounts to have fillers for all open boxes. We refine Kan's notion in two aspects: first, we require these fillers to be explicitly given as operations, and, second, that these operations satisfy certain uniformity conditions which ensure that the filling operations commute with the name substitutions in a suitable way. The second refinement is crucially used to show that types are closed under dependent function spaces.

This licentiate thesis presents this model for type theory with identity types, Σ-types, Π-types, and a universe. The interpretation of the identity type, however, only satisfies the usual equation of the J-eliminator up to propositional equality and *not* as definitional equality as is usually required in type theory. The thesis is based on the publication [7] and adds more detailed proofs. A new contribution in this thesis is the semantics of universes as Kan cubical sets. The treatment of the Univalence Axiom is not included here, but will be part of a future publication; sketches of the verification of univalence in this model are however given in [7, Section 8.4] and [13].

Moreover, this model (in its nominal set presentation) has been, implemented as the type-checker Cubical [11][2] together with C. Cohen, T. Coquand, and A. Mörtberg. This implementation builds on top of ordinary dependent type theory (without identity types) where certain primitive constants (giving rise to the properties we want for propositional equality) are available; whenever the type checker requires to check for conversion of two terms (i.e., definitional equality) we compare their semantics in the model. The implementation supports computing with the Univalence Axiom and in particular transporting along an equivalence.

Outline

In Chapter 1 we define semantics of Martin-Löf type theory relying on the notion of categories with families; we show how presheaves form an instance of this structure and thus give rise to a model of type theory. Chapter 2 introduces cubical sets as presheaves on the so-called cubical category along with examples. We also look into the relationship to nominal sets. Chapter 3 is the heart of this thesis: we introduce the notion of (uniform) Kan cubical sets and show that these induce a category with families extending the presheaf semantics. We give the interpretation of dependent sums, dependent products, and identity types. We also show how any cubical set can be "completed" to a Kan cubical set. In Chapter 4 we give the construction of a universe of Kan cubical set and show that it is itself a Kan cubical set, thus providing the semantics of a type theoretic universe in our model.

[2]The version relevant for this thesis is on the `master` branch dated September 24, 2014.

Chapter 1

Semantics of Martin-Löf Type Theory

In this chapter we will introduce semantics of type theory based on the notion of categories with families. As an extended example we explain how presheaf categories induce such a model of type theory. This is the basis for the model described in the next chapters.

1.1 Categories with Families

There are various similar notions to organize models of dependent type theory. In what follows we chose categories with families (CwF) which were introduced by Dybjer [17] and further popularized by Hofmann [25]. Categories with families can be seen as an *algebraic* presentation of type theory. Even though the definition of a CwF is given using categorical language we want to stress the fact that it is an instance of a generalized algebraic theory [9]. To devise a CwF is to give: interpretations (as sets) for the sorts of *contexts*, *context morphisms*, *types*, and *terms*; operations including the context extension; and to check equations involving those operations.

We will not be concerned with the (non-trivial) task to interpret the syntax of Martin-Löf type theory into a CwF. We refer the reader to [25] for a sketch of this.

The category of families of sets, **Fam**, has as objects (A, B) where A is a set and $B = (B_a \mid a \in A)$ is a A-indexed family of sets B_a, $a \in A$. A morphism between $(A, B) \to (A', B')$ is given by a pair (f, g) where $f \colon A \to A'$ is a function and g is an A-indexed family of functions such that $g_a \colon B_a \to B'_{fa}$.

Definition 1.1. A *category with families* (CwF for short) is given by $(\mathcal{C}, \mathcal{F})$ where:

1. $\mathcal{C}$ is a category whose objects $\Gamma, \Delta, \dots$ we call *contexts* and whose morphisms $\sigma, \tau, \dots$ we call *substitutions* or *context morphisms*; we write $\Gamma \vdash$

to indicate that Γ is an object of $\mathcal{C}$, if $\mathcal{C}$ is clear from the context.

2. A terminal object 1 in $\mathcal{C}$ called the *empty context.*

3. $\mathcal{F}$ is a functor $\mathcal{F}\colon \mathcal{C}^{\mathrm{op}} \to \mathbf{Fam}$; for $\Gamma \vdash$ we write $\mathrm{Ty}_{\mathcal{F}}(\Gamma)$ for the indexing set of the family $\mathcal{F}(\Gamma)$ and its family as $(\mathrm{Ter}_{\mathcal{F}}(\Gamma; A) \mid A \in \mathrm{Ty}_{\mathcal{F}}(\Gamma))$; we also write $\Gamma \vdash A$ for $A \in \mathrm{Ty}_{\mathcal{F}}(\Gamma)$ and call A a *type in context* Γ (or over Γ), and $\Gamma \vdash a : A$ for $a \in \mathrm{Ter}_{\mathcal{F}}(\Gamma; A)$ and call a a term of type A in context Γ. For a substitution $\sigma\colon \Delta \to \Gamma$ the morphism $\mathcal{F}(\sigma)$ acts on types $\Gamma \vdash A$ as $A\sigma$, and on terms $\Gamma \vdash a : A$ as $a\sigma$. Note that $\Delta \vdash A\sigma$ and $\Delta \vdash a\sigma : A\sigma$, and the fact that $\mathcal{F}$ is a functor yields:

$$A\mathbf{1} = A \quad (A\sigma)\tau = A(\sigma\tau) \quad a\mathbf{1} = a \quad (a\sigma)\tau = a(\sigma\tau)$$

(Here and henceforth $\mathbf{1}$ denotes the suitable identity morphism.)

4. The operations of *context extension*: if $\Gamma \vdash$ and $\Gamma \vdash A$, there is a context $\Gamma.A$, a context morphism $\mathsf{p}\colon \Gamma.A \to \Gamma$, and a term $\Gamma.A \vdash \mathsf{q} : A\,\mathsf{p}$. This should satisfy the following (universal) property: for each $\Delta \vdash$ and substitution $\sigma\colon \Delta \to \Gamma$ and $\Delta \vdash a : A\sigma$, there is a substitution $(\sigma, a)\colon \Delta \to \Gamma.A$ such that:

$$\mathsf{p}(\sigma, a) = \sigma \qquad \mathsf{q}(\sigma, a) = a \qquad (\sigma, a)\tau = (\sigma\tau, a\tau) \qquad (\mathsf{p}, \mathsf{q}) = \mathbf{1}$$

where in the last equation $\mathbf{1}\colon \Gamma.A \to \Gamma.A$.

Note that above we use polymorphic notation to increase readability as in [9, 17]; e.g., without this convention we should have written $\mathsf{p}_{\Gamma,A}$ for the first projection $\mathsf{p}\colon \Gamma.A \to \Gamma$. We also leave parameters implicit, so, e.g., in the equation $A\sigma\tau = A(\sigma\tau)$ tacitly assumes premises $\sigma\colon \Delta \to \Gamma$, $\tau\colon \Theta \to \Delta$, and $\Gamma \vdash A$.

Example 1.2. We can make the category of sets **Set** into the category of contexts of a CwF if we set the types over a set Γ to be the families of (small) sets A_γ indexed over $\gamma \in \Gamma$. A term is simply a dependent function $a_\gamma \in A_\gamma$ for each $\gamma \in \Gamma$, i.e., an element in the (set-theoretic) dependent function space $\Pi_{\gamma\in\Gamma} A_\gamma$. Context extensions are defined by the disjoint union

$$\Gamma.A = \{(\gamma, a) \mid \gamma \in \Gamma \text{ and } a \in A_\gamma\}$$

with $\mathsf{p}(\gamma, a) = a$ and $\mathsf{q}(\gamma, a) = a$.

As already indicated in the notation above we will usually suppress the reference to the CwF if clear from context. Moreover, we will usually present properties in rule form from now on. So, e.g., we will present the above definition in rule form as:

$$\frac{}{1 \vdash} \qquad \frac{\Gamma \vdash \qquad \Gamma \vdash A}{\Gamma.A \vdash}$$

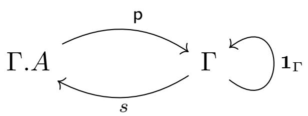
$$\frac{\Gamma \vdash A \qquad \sigma : \Delta \to \Gamma}{\Delta \vdash A\sigma} \qquad \frac{\Gamma \vdash a : A \qquad \sigma : \Delta \to \Gamma}{\Delta \vdash a\sigma : A\sigma}$$

$$\frac{\Gamma \vdash A}{\mathsf{p} : \Gamma.A \to \Gamma} \qquad \frac{\Gamma \vdash A}{\Gamma.A \vdash \mathsf{q} : A\mathsf{p}}$$

$$\frac{\sigma : \Delta \to \Gamma \qquad \Gamma \vdash A \qquad \Delta \vdash a : A\sigma}{(\sigma, a) \colon \Delta \to \Gamma.A}$$

If we have a type $\Gamma \vdash A$ and a dependent type or term over A, say $\Gamma.A \vdash B$, the operation to substitute a term of A, say $\Gamma \vdash a \colon A$, is represented in a CwF as follows. We have the identity context morphism $\mathbf{1} \colon \Gamma \to \Gamma$ and hence we can form $(\mathbf{1}, a) \colon \Gamma \to \Gamma.A$ which we usually denote by $[a] \colon \Gamma \to \Gamma.A$; this gives a type $\Gamma \vdash B[a]$.

Remark 1.3. Terms $\Gamma \vdash a : A$ are in a one-to-one correspondence with sections $s \colon \Gamma \to \Gamma.A$ of $\mathsf{p} : \Gamma.A \to \Gamma$, i.e., $\mathsf{p}s = \mathbf{1}$,



and, moreover, all sections are of the form $[a]$ for some $\Gamma \vdash a : A$.

As the definition of CwF is an instance of a generalized algebraic theory [17, Section 2.2], there is a notion of morphism of CwFs: we have to give operators for each of the different sorts preserving the required equations. We will however not make use of this notion and refer the reader to [9, Section 11] for the precise definition.

A mere CwF does not give much structure and only models type dependencies. We are interested in CwFs that have more structure.

A CwF *supports* Π-*types* if it is closed under the following rules:

$$\frac{\Gamma \vdash A \qquad \Gamma.A \vdash B}{\Gamma \vdash \Pi A B} \qquad \frac{\Gamma.A \vdash b : B}{\Gamma \vdash \lambda\, b : \Pi A B} \qquad \frac{\Gamma \vdash u : \Pi A B \qquad \Gamma \vdash a : A}{\Gamma \vdash \mathsf{app}(u, a) : B[a]}$$

Furthermore, we require the operations to satisfy β- and η-laws

$$\mathsf{app}(\lambda\, b, a) = b[a]$$
$$u = \lambda\ \mathsf{app}(u\mathsf{p}, \mathsf{q})$$

and laws for commutation with substitutions:

$$(\Pi A B)\sigma = \Pi(A\sigma)(B(\sigma\mathsf{p},\mathsf{q}))$$
$$(\lambda\, b)\sigma = \lambda\,(b(\sigma\mathsf{p},\mathsf{q}))$$
$$\mathsf{app}(u,a)\sigma = \mathsf{app}(u\sigma, a\sigma)$$

Note that these equations only make sense in the appropriate types. For, e.g., $(\lambda\, b)\sigma = \lambda\,(b(\sigma\mathsf{p},\mathsf{q}))$ to make sense, we need the equation $(\Pi A B)\sigma = \Pi(A\sigma)(B(\sigma\mathsf{p},\mathsf{q}))$.

Likewise, a CwF *supports Σ-types* if it is closed under the following rules:

$$\frac{\Gamma \vdash A \qquad \Gamma.A \vdash B}{\Gamma \vdash \Sigma A B} \qquad \frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B[a]}{\Gamma \vdash (a,b) : \Sigma A B}$$

$$\frac{\Gamma \vdash u : \Sigma A B}{\Gamma \vdash \mathsf{p}u : A} \qquad \frac{\Gamma \vdash u : \Sigma A B}{\Gamma \vdash \mathsf{q}u : B[\mathsf{p}u]}$$

Note that we overload the notation for pairs and projections in Σ-types with the notation for context morphisms as they are required to satisfy similar equations:

$$\mathsf{p}(a,b) = a$$
$$\mathsf{q}(a,b) = b$$
$$u = (\mathsf{p}u, \mathsf{q}u)$$

and the laws for substitutions:

$$(\Sigma A B)\sigma = \Sigma(A\sigma)(B(\sigma\mathsf{p},\mathsf{q}))$$
$$(a,b)\sigma = (a\sigma, b\sigma)$$
$$(\mathsf{p}u)\sigma = \mathsf{p}(u\sigma)$$
$$(\mathsf{q}u)\sigma = \mathsf{q}(u\sigma)$$

Note that the above definitions for a CwF to support a type former are rather direct from the corresponding syntactical formulation. A morphism of CwFs between CwFs supporting a certain structure (like Π-types) preserves this structure if the type and term formers are preserved.

Since it is slightly easier to use later on, we use Paulin-Mohring's formulation of the identity type [35]. A CwF *supports identity types* if it is closed under the following rules:

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash a : A \qquad \Gamma \vdash b : A}{\Gamma \vdash \mathtt{Id}_A(a,b)} \qquad \frac{\Gamma \vdash a : A}{\Gamma \vdash \mathtt{refl}\, a : \mathtt{Id}_A(a,a)}$$

$$\frac{\begin{array}{c}\Gamma \vdash A \qquad \Gamma \vdash a : A \qquad \Gamma.A.\,\mathtt{Id}_{A\mathsf{p}}(a\mathsf{p},\mathsf{q}) \vdash C \\ \Gamma \vdash v : C[a, \mathtt{refl}\, a] \qquad \Gamma \vdash b : A \qquad \Gamma \vdash p : \mathtt{Id}_A(a,b)\end{array}}{\Gamma \vdash \mathsf{J}(a,v,b,p) : C[b,p]}$$

Where $[b, p]$ is the substitution $([b], p)\colon \Gamma \to \Gamma.A.\,\mathtt{Id}_{A\mathsf{p}}(a\mathsf{p}, \mathsf{q})$. As before, we require that the operations commute with substitution:

$$(\mathtt{Id}_A(a,b))\sigma = \mathtt{Id}_{A\sigma}(a\sigma, b\sigma)$$
$$(\mathtt{refl}\,a)\sigma = \mathtt{refl}(a\sigma)$$
$$(\mathtt{J}(a,v,b,p))\sigma = \mathtt{J}(a\sigma, v\sigma, b\sigma, p\sigma)$$

where in the last equation $\mathtt{J}$ on the right hand side is w.r.t. C, and on the left hand side w.r.t. $C(\sigma\mathsf{p}, \mathsf{q})$. (Note that $\mathtt{J}$ depends on C although suppressed in our syntax.) Additionally, we require

$$\mathtt{J}(a, v, a, \mathtt{refl}\,a) = v. \tag{1.1}$$

The identity type of the model we consider later in Chapter 3 will not satisfy equation (1.1). For this reason we say that a CwF supports *weak identity types* if all conditions of identity types except equation (1.1) are satisfied but where equation (1.1) holds only propositionally, i.e., only up to a witness of a respective identity type, that is, we require the rule

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash a : A \qquad \Gamma.A.\,\mathtt{Id}_{A\mathsf{p}}(a\mathsf{p}, \mathsf{q}) \vdash C \qquad \Gamma \vdash v : C[a, \mathtt{refl}\,a]}{\Gamma \vdash \mathtt{JEq}(a, v) : \mathtt{Id}_{C[a,\mathtt{refl}\,a]}(v, \mathtt{J}(a, v, a, \mathtt{refl}\,a))}$$

such that $(\mathtt{JEq}(a, v))\sigma = \mathtt{JEq}(a\sigma, v\sigma)$.

There are usually two different ways to introduce a universe to dependent type theory: universes à la Tarski and universes à la Russel. A universe à la Tarski contains *codes* of the actual types and is equipped with a function decoding codes into types, where as a universe à la Russel has *types* as its elements. We will use a variant of a universe à la Tarski but in a formulation which is less general but simpler and is sufficient for the universes we consider (cf. Section 1.2.4).

Definition 1.4. A *universe* in a CwF $(\mathcal{C}, \mathcal{F})$ is a CwF $\mathcal{U} = (\mathcal{C}, \mathcal{F}_0)$ on the same category of contexts and substitutions such that $\mathrm{Ty}_{\mathcal{F}_0}(\Gamma) \subseteq \mathrm{Ty}_{\mathcal{F}}(\Gamma)$ and $\mathrm{Ter}_{\mathcal{F}_0}(\Gamma; A) = \mathrm{Ter}_{\mathcal{F}}(\Gamma; A)$ if $A \in \mathrm{Ty}_{\mathcal{F}_0}(\Gamma)$, and the context operations of $\mathcal{U}$ are inherited from $(\mathcal{C}, \mathcal{F})$. We write $\Gamma \vdash A\ \mathsf{Type}_0$ for $A \in \mathrm{Ty}_{\mathcal{F}_0}(\Gamma)$ and call A a *$\mathcal{U}$-small type*; thus we require:

$$\frac{\Gamma \vdash A\ \mathsf{Type}_0}{\Gamma \vdash A}$$

Moreover, we require that there is a type $1 \vdash U$ (writing also U for $U\sigma$ for the (unique) substitution $\sigma\colon \Gamma \to 1$) equipped with coding and decoding functions

$$\frac{}{1 \vdash U} \qquad \frac{\Gamma \vdash A\ \mathsf{Type}_0}{\Gamma \vdash \ulcorner A \urcorner : U} \qquad \frac{\Gamma \vdash T : U}{\Gamma \vdash \mathrm{El}\,T\ \mathsf{Type}_0}$$

$$U\sigma = U \qquad \ulcorner A \urcorner \sigma = \ulcorner A\sigma \urcorner \qquad (\mathrm{El}\,T)\sigma = \mathrm{El}(T\sigma)$$

satisfying the equations

$$\mathrm{El}\ulcorner A \urcorner = A \quad \text{and} \quad \ulcorner \mathrm{El}\, T \urcorner = T. \tag{1.2}$$

If $(\mathcal{C}, \mathcal{F})$ supports Π-types, we say that the universe U supports Π-types if $(\mathcal{C}, \mathcal{F}_0)$ is closed under the induced Π-types, or in other words if U supports Π-types and the inclusion CwF-morphism $\mathcal{U} \hookrightarrow (\mathcal{C}, \mathcal{F})$ preserves Π. Similarly for other type formers.

Note with the equations (1.2) there is no need to require coding functions for the type formers which simplifies the treatment of universes significantly. E.g., for Π-types given $\Gamma \vdash a : U$ and $\Gamma.\, \mathrm{El}\, a \vdash b : U$ we can define $\Gamma \vdash \pi\ a\ b : U$ by

$$\pi\ a\ b = \ulcorner \Pi(\mathrm{El}\, a)(\mathrm{El}\, b) \urcorner$$

which satisfies $\mathrm{El}(\pi\ a\ b) = \Pi(\mathrm{El}\, a)(\mathrm{El}\, b)$.

1.2 Presheaf Models of Type Theory

We will now show how any presheaf category gives rise to a category with families where the contexts are presheaves. Let us first recall the notion of presheaf.

Definition 1.5. Let $\mathcal{C}$ be a category. A *presheaf* on $\mathcal{C}$ is a contravariant functor from $\mathcal{C}$ into **Set**. The category of presheaves on $\mathcal{C}$, denoted by $\mathrm{Psh}(\mathcal{C})$ or sometimes $\hat{\mathcal{C}}$, is the functor category $[\mathcal{C}^{\mathrm{op}}, \mathbf{Set}]$. In particular its morphisms are natural transformations.

Let us now fix a small category $\mathcal{C}$. In this section, we denote objects of $\mathcal{C}$ by I, J, K and morphisms by f, g, h. In what follows we describe how $\mathrm{Psh}(\mathcal{C})$ induces a CwF where the category of contexts is $\mathrm{Psh}(\mathcal{C})$.

So a context $\Gamma \vdash$ is a presheaf Γ on $\mathcal{C}$ is a functor $\Gamma \colon \mathcal{C}^{\mathrm{op}} \to \mathbf{Set}$, i.e., we are given a set $\Gamma(I)$ for each I in $\mathcal{C}$, and functions $\Gamma(I) \to \Gamma(J), \rho \mapsto \rho f$ (called *restriction*, and written as f acting on the right) for each $f \colon J \to I$, such that

$$\rho \mathbf{1} = \rho \quad \text{and} \quad (\rho f) g = \rho (f g)$$

for $g \colon K \to J$ and $f \colon J \to I$. In this section we will sometimes refer to the map $\rho \mapsto \rho f$ as Γ_f.

The empty context 1 is the terminal presheaf which is constant a singleton $\{\star\}$.

A context morphism σ between contexts Γ and Δ is then a natural transformation $\sigma \colon \Delta \to \Gamma$, i.e., for each I in $\mathcal{C}$ there is a map $\sigma_I \colon \Delta(I) \to \Gamma(I)$ such that for any $f \colon J \to I$ the square

$$\begin{array}{ccc} \Delta(I) & \xrightarrow{\sigma_I} & \Gamma(I) \\ {\scriptstyle \Delta_f}\downarrow & & \downarrow{\scriptstyle \Gamma_f} \\ \Delta(J) & \xrightarrow{\sigma_J} & \Gamma(J) \end{array}$$

commutes, i.e., $\Gamma_f \circ \sigma_I = \sigma_J \circ \Delta_f$. From now on we will suppress writing the subscripts to σ; this way, if we write Γ_f and Δ_f as f acting on the right again, the equation simply becomes

$$(\sigma\rho)f = \sigma(\rho f)$$

for ρ in $\Delta(I)$.

Next, we describe how to give a dependent type $\Gamma \vdash A$ in a context $\Gamma \vdash$. For each object $I \in \mathcal{C}$ and $\rho \in \Gamma(I)$ we require a set $A\rho$, and for each $f \colon J \to I$, we require a function $A\rho \to A(\rho f)$, written as $a \mapsto af$ satisfying $a\mathbf{1} = a$ and $afg = a(fg)$ if $g \colon K \to J$. Note that we tacitly suppressed the dependence on I in $A\rho$ in our notation to keep it lighter; similarly we omit I and ρ in af. Substitution $\Delta \vdash A\sigma$ with $\sigma \colon \Delta \to \Gamma$ is simply given $(A\sigma)\rho = A(\sigma\rho)$ for $\rho \in \Delta(I)$, together with the induced map

$$(A\sigma)\rho = A(\sigma\rho) \to A((\sigma\rho)f) = A(\sigma(\rho f)) = (A\sigma)(\rho f)$$

for $f \colon J \to I$. Clearly, this satisfies the required equations for substitutions.

Note that types in the empty context $1 \vdash A$ correspond exactly to contexts $A \vdash$. We will usually write $\vdash A$ instead of $1 \vdash A$.

The definition of a dependent type can also be rephrased more categorically: A is a presheaf on $\int_{\mathcal{C}} \Gamma$, where $\int_{\mathcal{C}} \Gamma$ is the *category of elements* of the presheaf Γ, defined as follows:

- objects are pairs (I, ρ) where $I \in \mathcal{C}$ and $\rho \in \Gamma(I)$;
- a morphism $(J, \rho') \to (I, \rho)$ is a morphism $f \colon J \to I$ in $\mathcal{C}$ such that $\rho' = \Gamma_f \rho$.

Note that substitution of $A \colon (\int_{\mathcal{C}} \Gamma)^{\mathrm{op}} \to \mathbf{Set}$ with $\sigma \colon \Delta \to \Gamma$ corresponds to precomposing with $\int_{\mathcal{C}} \sigma \colon \int_{\mathcal{C}} \Delta \to \int_{\mathcal{C}} \Gamma$ induced by σ. (In fact, this construction induces a functor $\int_{\mathcal{C}} \colon \mathrm{Psh}(\mathcal{C}) \to \mathbf{Cat}$.)

A term $\Gamma \vdash a : A$ of a dependent type $\Gamma \vdash A$ is given by a family of elements $a\rho \in A\rho$ for each I in $\mathcal{C}$ and $\rho \in \Gamma(I)$, such that $a\rho f = a(\rho f)$ for each $f \colon J \to I$. The substitution $\Delta \vdash a\sigma : A\sigma$ with $\sigma \colon \Delta \to \Gamma$ is given by the family $(a\sigma)\rho = a(\sigma\rho)$.

For $\Gamma \vdash A$, the context extension $\Gamma.A \vdash$ is defined by

$$\begin{aligned} (\Gamma.A)(I) &= \{(\rho, u) \mid \rho \in \Gamma(I) \text{ and } u \in A\rho\} && \text{for } I \in \mathcal{C} \\ (\rho, u)f &= (\rho f, uf) && \text{for } f \colon J \to I \end{aligned}$$

and the projections are defined by $\mathsf{p} \colon \Gamma.A \to \Gamma$, $\mathsf{p}(\rho, u) = \rho$, and $\Gamma.A \vdash \mathsf{q} : A\mathsf{p}$ by $\mathsf{q}(\rho, u) = u$. Now assume $\sigma \colon \Delta \to \Gamma$ and $\Delta \vdash a : A\sigma$; we define $(\sigma, a) \colon \Delta \to \Gamma.A$ by $(\sigma, a)\rho = (\sigma\rho, a\rho)$. One readily checks that this satisfies all the required equations; this concludes the definition of the CwF associated to a presheaf category.

For later use let us recall the Yoneda Lemma: the Yoneda embedding is the functor $\mathbf{y}\colon \mathcal{C} \to \hat{\mathcal{C}}$ is given by $\mathbf{y}I = \mathrm{Hom}_{\mathcal{C}}(-, I)$, i.e.,

$$(\mathbf{y}I)(J) = \mathrm{Hom}_{\mathcal{C}}(J, I) \quad \text{for } I \text{ an object of } \mathcal{C}$$
$$(\mathbf{y}I)_f\colon (\mathbf{y}I)(J) \to (\mathbf{y}I)(K), g \mapsto fg \quad \text{for } f\colon K \to J \text{ in } \mathcal{C}.$$

The Yoneda functor is fully faithful and we have for a presheaf Γ,

$$\Gamma(I) \cong \mathrm{Hom}_{\mathrm{Psh}(\mathcal{C})}(\mathbf{y}I, \Gamma)$$

both natural in Γ and I. A presheaf Γ is representable if it is naturally isomorphic to a $\mathbf{y}I$ for some I.

1.2.1 Dependent Products

As a motivation for the definition of dependent products let us recall how to define exponents in presheaf categories. Let Γ and Δ be presheaves and suppose we already know how the exponent Δ^Γ is constructed. Then we get by the Yoneda Lemma and the fact that $-^\Gamma$ should be right adjoint to $- \times \Gamma$, that

$$\begin{aligned}(\Delta^\Gamma)(I) &\cong \mathrm{Hom}(\mathbf{y}I, \Delta^\Gamma)\\ &\cong \mathrm{Hom}(\mathbf{y}I \times \Gamma, \Delta)\end{aligned}$$

The latter can now be taken *as a definition*. So an element w of $(\Delta^\Gamma)(I)$ is a natural transformation $w\colon \mathbf{y}I \times \Gamma \to \Delta$, so it is given by functions

$$w_J\colon (\mathbf{y}I)(J) \times \Gamma(J) \to \Delta(J),$$

and the naturality condition becomes $(w_J(f, \rho))g = w_K(fg, \rho g)$ for $f\colon J \to I$, $\rho \in \Gamma(J)$, and $g\colon K \to J$.

We will now show that the CwF associated to a presheaf category supports Π-types. Given $\Gamma \vdash A$ and $\Gamma.A \vdash B$ we have to define $\Gamma \vdash \Pi AB$, that is, we have to define the set $(\Pi AB)\rho$ for $I \in \mathcal{C}$ and $\rho \in \Gamma(I)$. The elements w of $(\Pi AB)\rho$ are families $w = (w_f \mid J \in \mathcal{C}, f\colon J \to I)$ of (dependent) functions such that

$$w_f u \in B(\rho f, u) \quad \text{for } J \in \mathcal{C}, f\colon J \to I, \text{ and } u \in A(\rho f),$$

with the requirement that for $g\colon K \to J$

$$(w_f u)g = w_{fg}(ug).$$

For such a family $w \in (\Pi AB)\rho$, the restriction $wf \in (\Pi AB)(\rho f)$ for $f\colon J \to I$ is defined by taking

$$(wf)_g u = w_{fg} u \in B(\rho f g, u)$$

where $g\colon K \to J$ and $u \in A(\rho f g)$. (Note that this needs $\rho f g = \rho(fg)$.) This definition satisfies $w\mathbf{1} = w$ and $wfg = w(fg)$.

Now let $\Gamma.A \vdash b : B$; we have to define $\Gamma \vdash \lambda\, b : \Pi AB$, that is, give $(\lambda\, b)\rho \in (\Pi AB)\rho$. For $f\colon J \to I$ and $u \in A(\rho f)$ we set

$$((\lambda\, b)\rho)_f u = b(\rho f, u) \in B(\rho f, u).$$

This satisfies for $g\colon K \to J$

$$(((\lambda\, b)\rho)_f u)g = (b(\rho f, u))g = b(\rho(fg), ug) = ((\lambda\, b)\rho)_{fg}(ug)$$

and defines a term since

$$(((\lambda\, b)\rho)f)_g u = ((\lambda\, b)\rho)_{fg} u = b(\rho f g, u) = ((\lambda\, b)(\rho f))_g u.$$

To define the application let $\Gamma \vdash u : \Pi AB$ and $\Gamma \vdash v : A$, and we set $\mathtt{app}(u, v)\rho = (u\rho)_{\mathbf{1}}(v\rho) \in B(\rho, v\rho)$, thus $\mathtt{app}(u, v)\rho \in B[v]\rho$ as $(\rho, v\rho) = (\mathbf{1}, v)\rho = [v]\rho$. β-equality is readily checked

$$\mathtt{app}(\lambda\, b, v)\rho = ((\lambda\, b)\rho)_{\mathbf{1}}(v\rho) = b(\rho, v\rho) = b[v]\rho,$$

and similarly for η-equality and the other equations.

It is also possible to calculate the dependent product using the Yoneda Lemma similar to exponents (i.e., non-dependent products); this does however not add much to the explanation so we have refrained from adding it.

Remark 1.6. It is worthwhile to note that there is a simpler way to describe the sections of Π-types. Given a section $\Gamma \vdash w : \Pi AB$ it satisfies $(w\rho)_f = ((w\rho)f)_{\mathbf{1}} = (w(\rho f))_{\mathbf{1}}$ by definition. This entails that w is determined by the $(w\rho)_{\mathbf{1}}$'s and moreover we have

$$((w\rho)_{\mathbf{1}} a)f = (w\rho)_f(af) = (w(\rho f))_{\mathbf{1}}(af).$$

Conversely, assume that we have a family φ of functions $\varphi\rho$ such that $\varphi\rho\, a \in B(\rho, a)$ for $a \in A\rho$ satisfying

$$(\varphi\rho\, a)f = \varphi(\rho f)(af).$$

This defines a section $\Gamma \vdash v : \Pi AB$ by putting

$$(v\rho)_f a = \varphi(\rho f)a.$$

These assignments are inverse to each other. Using this representation, application can be simply written as

$$\mathtt{app}(\varphi, v)\rho = \varphi\rho(v\rho),$$

and abstraction as $(\lambda\, u)\rho\, a = u(\rho, a)$.

1.2.2 Dependent Sums

The interpretation Σ-types $\Gamma \vdash \Sigma AB$ for $\Gamma \vdash A$ and $\Gamma.A \vdash B$ is defined by

$$(\Sigma AB)\rho = \{(a,b) \mid a \in A\rho \text{ and } b \in B(\rho,a)\}$$

for $\rho \in \Gamma(I)$, $I \in \mathcal{C}$. The restrictions are defined componentwise $(a,b)f = (af, bf)$ for $f\colon J \to I$. The pairing operation $\Gamma \vdash (u,v) : \Sigma AB$ of $\Gamma \vdash u : A$ and $\Gamma.A \vdash v : B[u]$ is defined by the componentwise pairing: $(u,v)\rho = (u\rho, v\rho)$. Likewise for the projections p and q, $(\mathsf{p}w)\rho = a$ and $(\mathsf{q}w)\rho = b$ for $w\rho = (a,b)$. This validates the necessary equations.

1.2.3 Identity Types

There is also a "standard" interpretation for identity types in a presheaf model. However, this is *not* the interpretation we are interested in since it is proof irrelevant and satisfies uniqueness of identity proofs. Thus we will only briefly sketch the definition: for $\Gamma \vdash A$, $\Gamma \vdash a : A$, and $\Gamma \vdash b : A$, the identity type is given by $(\mathtt{Id}_A(a,b))\rho = \{* \mid a\rho = b\rho\}$. This allows an interpretation for the rules of identity types, e.g., $(\mathtt{refl}\, a)\rho = *$; this interpretation is such that the equation (1.1) on page 11 holds. In the rest of this thesis, we will not make further use of this identity type.

1.2.4 Universes

We will now show how to lift a universe à la Grothendieck in the underlying set theory to the presheaf model following [26, 40].

Assume a universe of small sets, say $\mathbf{Set}_0 \in \mathbf{Set}$, and call the elements small sets. From this we will now give a type theoretic universe. We recall from Definition 1.4 that to give a type theoretic universe we have to first single out the small types: the judgment $\Gamma \vdash A\ \mathsf{Type}_0$ is defined to mean that for each $\rho \in \Gamma(I)$, the set $A\rho$ is small, i.e., $A\rho \in \mathbf{Set}_0$. In this case we call A a small type (in context Γ). We clearly have that any small type is a type, i.e.,

$$\frac{\Gamma \vdash A\ \mathsf{Type}_0}{\Gamma \vdash A}$$

Since our underlying universe of small sets $\mathbf{Set}_0$ is closed under set-theoretic operations we get that the small types $\Gamma \vdash A\ \mathsf{Type}_0$ form itself a CwF of small types supporting the discussed type forming operations like Π, Σ, and are closed under substitution.

Next, we have to give the type of codes for small types in the empty context $\vdash U$. This is the same as giving a context $U \vdash$. The definition has to be such that for each small type $\Gamma \vdash A\ \mathsf{Type}_0$ there is a code $\Gamma \vdash \ulcorner A \urcorner : U$, and for each section of U, say $\Gamma \vdash T : U$, there is a small type $\Gamma \vdash \mathrm{El}\, T\ \mathsf{Type}_0$ satisfying:

$$\mathrm{El} \ulcorner A \urcorner = A \qquad\qquad \ulcorner \mathrm{El}\, T \urcorner = T$$

We now define U as a context. For $I \in \mathcal{C}$, $U(I)$ consists of *small* types

$$\mathbf{y}I \vdash A \ \mathsf{Type}_0$$

where $\mathbf{y}$ denotes the Yoneda embedding. The restrictions are as well given by the Yoneda embedding: if $f\colon J \to I$ and $A \in U(I)$, that is, $\mathbf{y}I \vdash A$ is small, then the restriction $Af \in U(J)$ is defined to be the small type $\mathbf{y}J \vdash A\mathbf{y}f$, where $\mathbf{y}f : \mathbf{y}J \to \mathbf{y}I$.

Let us unfold the above definition: for $I \in \mathcal{C}$, $U(I)$ consists of small types $\mathbf{y}I \vdash A \ \mathsf{Type}_0$, that is, for each f with $\operatorname{cod} f = I$ a small set A_f (we use a subscript to not confuse the set A_f with the restriction of A along f written as Af), and A comes together with maps $A_f \to A_{fg}$, $a \mapsto ag$ for $g\colon K \to J$, $f\colon J \to I$ such that $a\mathbf{1} = a$ and $agh = a(gh)$ for $h\colon L \to K$. The restriction $U(I) \to U(J)$ along $f\colon J \to I$ is $(Af)_g = A_{\mathbf{y}fg} = A_{fg}$ for $g\colon K \to J$.

In other words, the elements of $U(I)$ can also be described as $\mathbf{Set}_0$-valued presheaves on $\mathcal{C}/I$ (the slice category over I), i.e., $[(\mathcal{C}/I)^{\mathrm{op}}, \mathbf{Set}_0]$, with the restriction induced by $\mathcal{C}/f : \mathcal{C}/J \to \mathcal{C}/I$. This is reflected by the equivalence of categories:

$$\mathcal{C}/I \approx \int_{\mathcal{C}} \mathbf{y}I$$

Next, we define a small type $\Gamma \vdash \mathrm{El}\, T$ given $\Gamma \vdash T : U$. For $\rho \in U(I)$ we have $T\rho \in U(I)$ and set $(\mathrm{El}\, T)\rho = (T\rho)_{\mathbf{1}_I}$ which is a small set. The required map $(\mathrm{El}\, T)\rho \to (\mathrm{El}\, T)\rho f$ for $f\colon J \to I$ is defined to be the given map $(T\rho)_{\mathbf{1}} \to (T\rho)_f$, which makes sense since $(\mathrm{El}\, T)\rho f = (T\rho f)_{\mathbf{1}} = ((T\rho)f)_{\mathbf{1}_J} = (T\rho)_f$.

If $\sigma\colon \Delta \to \Gamma$ and $\Gamma \vdash T : U$, then $(\mathrm{El}\, T)\sigma = \mathrm{El}(T\sigma)$ since for $\rho \in \Delta(I)$

$$((\mathrm{El}\, T)\sigma)\rho = (\mathrm{El}\, T)(\sigma\rho) = (T(\sigma\rho))_{\mathbf{1}} = ((T\sigma)\rho)_{\mathbf{1}} = (\mathrm{El}(T\sigma))\rho.$$

Last, we define the code $\Gamma \vdash \ulcorner A \urcorner : U$ of a small type $\Gamma \vdash A \ \mathsf{Type}_0$. For $\rho \in \Gamma(I)$ we define $\ulcorner A \urcorner\rho \in U(I)$ as the small type $\mathbf{y}I \vdash \ulcorner A \urcorner\rho$ given by $(\ulcorner A \urcorner\rho)_f = A(\rho f)$ for $f \in (\mathbf{y}I)(J)$, i.e., $f\colon J \to I$; and the induced restriction maps

$$(\ulcorner A \urcorner\rho)_f = A(\rho f) \to A(\rho f)g = A(\rho(fg)) = (\ulcorner A \urcorner\rho)_{fg}.$$

The verification of $\ulcorner A \urcorner\sigma = \ulcorner A\sigma \urcorner$ for a substitution $\sigma\colon \Delta \to \Gamma$ is straightforward.

It remains to check the equation (1.2) of Definition 1.4: we have that

$$(\mathrm{El}\ulcorner A \urcorner)\rho = (\ulcorner A \urcorner\rho)_{\mathbf{1}} = A(\rho\mathbf{1}) = A\rho,$$

and likewise

$$(\ulcorner \mathrm{El}\, T \urcorner\rho)_f = (\mathrm{El}\, T)(\rho f) = (T(\rho f))_{\mathbf{1}} = (T\rho)_f.$$

Analogously, one can lift a hierarchy of Grothendieck universes $\mathbf{Set}_0 \in \mathbf{Set}_1 \in \cdots \in \mathbf{Set}_n \in \cdots \in \mathbf{Set}$ to a type theoretic hierarchy of universes $U_0, U_1, \ldots, U_n, \ldots$ with corresponding coding and decoding machinery.

Chapter 2

Cubical Sets

This chapter introduces cubical sets. Cubical sets are presheaves on the cubical category $\square$ introduced in Section 2.1.

2.1 The Cubical Category

We fix a countable and discrete set of atomic names $\mathbb{A}$ which will serve as explicit names for dimensions. We will denote elements of $\mathbb{A}$ by $x, y, z, \ldots$ and call them *names*. Later we will also assume that given a *finite* set of names $I \subseteq \mathbb{A}$ there is a fresh name x_I. The choice of the set $\mathbb{A}$ is irrelevant—what counts is that we can decide the equality of names. We will also assume that there are two elements 0 and 1 called *directions* which are *not* names; we will usually use c, d to denote directions; we write $\bar{c}$ for flipping the direction, i.e., $\bar{c} = 1 - c$. We set $\mathbf{2} = \{0, 1\}$.

Definition 2.1. The *cubical category* $\square$ has as objects finite subsets of the fixed set of names $\mathbb{A}$, usually denoted by $I, J, K, \ldots$. A morphisms $f \colon I \to J$ is given by a set-theoretic function $f \colon I \to J \cup \mathbf{2}$ such that for $x, y \in I$ with $f\,x, f\,y \notin \mathbf{2}$ we have

$$f\,x = f\,y \text{ implies } x = y,$$

i.e., f is injective when restricted to its *defined elements*

$$\operatorname{def}(f) = \{x \in I \mid f\,x \notin \mathbf{2}\}.$$

The composition of two morphism $f \colon I \to J$ and $g \colon J \to K$ is given by

$$(g \circ f)\,x = \begin{cases} g(fx) & \text{if } x \in \operatorname{def}(f), \\ fx & \text{otherwise.} \end{cases}$$

For the composition of two morphisms f and g as above we also write $fg = g \circ f$, i.e., we use *diagram order*. For finite sets of names we will write commas instead of unions and often omit curly braces; e.g., we write I, x, y for

$I \cup \{x, y\}$, and $I - x, y$ for the set-difference of I with $\{x, y\}$. From now on, if not stated otherwise, we assume that f, g, h range over morphisms in $\square$.

The *face maps* are morphisms $(x = 0), (x = 1) \colon I \to I - x$ for $x \in I$ sending x to 0 and 1, respectively; so in particular $\mathrm{def}\,(x = c) = I - x$. For $I \subseteq J$, we call the inclusion map $I \to J$ a *degeneracy map*; in particular, if $x \notin I$, we denote the inclusion $I \subseteq I, x$ by $s_x \colon I \to I, x$. For $x, y \notin I$ the *renaming* $(x = y) \colon I, x \to I, y$ sends x to y and leaves the rest untouched. (Note that it is crucial that both x and y are not in I, otherwise the injectivity condition on defined elements is violated.) For $x, y \in I$ the *swapping* of x and y, $(x\ y) \colon I \to I$, is defined by

$$(x\ y)z = \begin{cases} y & \text{if } z = x, \\ x & \text{if } z = y, \\ z & \text{otherwise.} \end{cases}$$

Note that with the standard programming trick to swap two variables using assignments and a third variable, we can write a swap $(x\ y) \colon I, x, y \to I, x, y$ (where $x, y \notin I$) as a composition of renamings: with a fresh z we have

$$\begin{array}{ccc} I, x, y & \xrightarrow{(x\ y)} & I, x, y \\ {\scriptstyle (x=z)}\big\downarrow & & \big\uparrow{\scriptstyle (z=y)} \\ I, y, z & \xrightarrow{(y=x)} & I, x, z \end{array}$$

For $f \colon I \to J$ set $f - x \colon I - x \to J - fx$ to be f restricted to $I - x$ with adapted codomain (where $J - fx = J$ if fx is 0 or 1). Similarly we can extend $f \colon I \to J$ for $x \notin I$ and a a name, 0, or 1, to $(f, x = a) \colon I, x \to J, a$ (by convention, $J, 0 = J, 1 = J$).

We call $f \colon I \to J$ *strict* if $\mathrm{def}(f) = I$.

The following lemma (whose proof is trivial) gives a factorization of any morphism as composition of faces, swapping (or renaming), and a degeneracy map.

Lemma 2.2. *Any $f \colon I \to J$ can be uniquely as $f = f_{01} g$ where $f_{01} \colon I \to \mathrm{def}(f)$ is a composition of face maps and $g \colon \mathrm{def}(f) \to J$ is a strict map. Moreover, if $I_0 = \vec{x} = f^{-1}(0)$ and $I_1 = \vec{y} = f^{-1}(1)$, then $f_{01} = (\vec{x} = 0)(\vec{y} = 1)$, and we have a commuting square*

$$\begin{array}{ccc} I & \xrightarrow{\quad f \quad} & J \\ {\scriptstyle f_{01}}\big\downarrow & \overset{g}{\nearrow} & \big\uparrow \\ \mathrm{def}(f) & \xrightarrow{\ f'\ } & \mathrm{im}(f) \end{array}$$

where f' is a bijection (and thus can be written a composition of transpositions, i.e., swappings, and thus also as a composition of renamings).

2.2 Cubical Sets

Similar to simplicial sets, cubical sets are defined as presheaves.

Definition 2.3. A *cubical set* X is a functor $X\colon \square \to \mathbf{Set}$, i.e., X is a presheaf over $\square^{\mathrm{op}}$.

That is, a cubical set X is given by sets $X(I)$ for each finite set of names I, and for each morphism in $f\colon I \to J$ in $\square$, a function $X(I) \to X(J)$, $u \mapsto uf$ such that

$$u\mathbf{1} = u \quad \text{and} \quad ufg = u(fg)$$

for $f\colon I \to J$ and $g\colon J \to K$ in $\square$. Note that the latter equation is the reason for the use of the diagram order for composition in $\square$; this enables us to write the action of a morphism on the right and still have the arrows from $\square$ and not its opposite category.

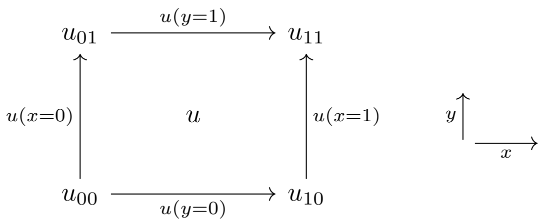
As with any presheaf category, the morphisms between cubical sets are given by natural transformations. This makes cubical sets into a category, denoted by **cSet**.

For $u \in X(I)$ and $x \in I$ we can form the faces $u(x = 0) \in X(I - x)$ and $u(x = 1) \in X(I - x)$ of u. In this way, we can think of u as a "line" connecting $u(x = 0)$ and $u(x = 1)$ along x, written as

$$u(x = 0) \xrightarrow[x]{u} u(x = 1) \tag{2.1}$$

We sometimes omit the subscript x in (2.1) if irrelevant or clear from the context.

In this way one can think of an element u in $X(I)$ as a hypercube of dimension $|I|$, and we call the elements of $X(I)$ also I*-cubes*. For example, if u in $X(x, y)$, $x \neq y$, first note that $u(x = c)(y = d) = u(y = d)(x = c)$ and thus with $u_{cd} = u(x = c)(y = d)$ we get a cube (indicating the naming of the dimensions on the right)



This cube should be thought as a "solid" cube, filled by u. Note that there is no "diagonal" in this cube, i.e., a face of u which connects u_{00} with u_{11}.

For an I-cube u and an $x \notin I$, we can consider the degenerate $us_x \in X(I, x)$ of u, connecting $us_x(x = 0) = u$ and $us_x(x = 1) = u$, i.e., u with itself along x:

$$u \xrightarrow[x]{us_x} u$$

Intuitively, one can think of us_x as the line which is constantly u. If $w = us_x$ for some u and x, we write that $w \# x$ borrowing notation from nominal sets (cf. Section 2.3) and say that w is *degenerate* (along x). Note that it is in general undecidable whether an element is degenerate.

Summing up, if we have an I-cube u we want to think of it as depending on the names in I; there are the following basic operations on u: renaming a name in I with a fresh name, setting one of the variables in I to 0 or 1, and adding a variable dependency (degeneracy maps).

From the fact that the category of cubical sets is a presheaf category we know that it has the structure of a topos and thus has a rich structure. In particular, we have seen in detail in Section 1.2 how **cSet** gives rise to a category with families. Let us introduce some examples of cubical sets.

Example 2.4. For every set A the *discrete cubical set* $\Delta(A)$ given by the constant presheaf $\Delta(A)(I) = A$ and $\Delta(A)(f) = \mathbf{1}_A$.

Example 2.5. A particularly natural example, suggested by Peter Aczel, is given by polynomial rings. Let R be a commutative ring with 1. For $I = x_1, \dots, x_n$ let $R[I] = R[x_1, \dots, x_n]$ denote the polynomial ring with indeterminates $x_1, \dots, x_n$ and coefficients in R. This assignment $I \mapsto R[I]$ defines a cubical set which we denote by $R[\cdot]$; if $f : I \to J$ and $p(x_1, \dots, x_n) \in R[I]$, pf is given by the polynomial $p(fx_1, \dots, fx_n)$. So, for example, $(1 + x^2y + z)(y = 0) = 1 + z$. Note that we assume $R[I] \subseteq R[I, x]$ and degeneracy is an inclusion: $ps_x = p$.

Example 2.6. The *interval* $\mathbb{I}$ is defined by $\mathbb{I}(I) = I \cup \mathbf{2}$ and $\mathbb{I}(f) : \mathbb{I}(I) \to \mathbb{I}(J)$ for $f : I \to J$ is defined by extending (the underlying set-theoretic map of) f with $\mathbb{I}(f)(0) = 0$ and $\mathbb{I}(f)(1) = 1$. Note that this is a representable cubical set: fix any name x, then $\mathbf{y}\{x\} \cong \mathbb{I}$. For example, $\mathbf{y}\{x\}\emptyset = \mathbf{cSet}(\{x\}, \emptyset)$ consists of the moprphisms $(x = 0), (x = 1) : x \to \emptyset$; similarly, $\mathbf{y}\{x\}\{y_1, \cdots, y_n\}$ consists of the morphisms $(x = a)$ for $a \in \mathbb{I}(y_1, \cdots, y_n)$.

More generally, we set $\square_I = \mathbf{y}I$ for a finite set of names I and call it the *standard I-cube*. For any I and J with n-elements, we clearly have $\square_I \cong \square_J$, and thus it makes sense to speak of $\square_I$ of a standard n-cube. The Yoneda Lemma gives that for a cubical set X, morphisms $\square_I \to X$ correspond to the elements of $X(I)$. Note that $\square_\emptyset$ is terminal in **cSet**, and thus $1 \to X$ corresponds to $X(\emptyset)$, the points of X.

Note that the product $\mathbb{I} \times \mathbb{I}$ is not isomorphic to $\square_{x,y}$. The latter does not contain the diagonal while the former does (cf. also the next example).

Example 2.7 (Nerve)**.** To any small category $\mathcal{C}$ we associate its *(cubical) nerve* $\mathrm{N}\mathcal{C}$ whose n-cubes are given by n-dimensional cubical commutative diagrams, defined as follows. For a finite set of names I we consider $\{0,1\}^I = \mathbf{2}^I$ as a poset, and hence category. The *set* $(\mathrm{N}\mathcal{C})(I)$ is defined to be the functors $\mathbf{2}^I \to \mathcal{C}$. Any $f : I \to J$ determines a monotone $\mathbf{2}^f : \mathbf{2}^J \to \mathbf{2}^I$, sending $\alpha \in \mathbf{2}^J$ to $(\mathbf{2}^f\alpha)(i) = \alpha(fi)$ for $i \in I$ and f defined on i, and $(\mathbf{2}^f\alpha)(i) = fi$ otherwise; for $\theta \in (\mathrm{N}\mathcal{C})(I)$, i.e., $\theta : \mathbf{2}^I \to \mathcal{C}$ a functor, we set the restriction θf to be $\theta\mathbf{2}^f : \mathbf{2}^J \to \mathbf{2}^I \to \mathcal{C}$.

Elements of $(\mathrm{N}\,\mathcal{C})(\emptyset)$ are functors $1 \to \mathcal{C}$, i.e., correspond to objects in $\mathcal{C}$; elements of $(\mathrm{N}\,\mathcal{C})(x)$ are functors $\alpha\colon \{0,1\} \to \mathcal{C}$, i.e., correspond to morphisms in $\mathcal{C}$ (given by $\alpha(0 \leq 1)$, $0 \leq 1$ denoting the (unique) arrow $0 \to 1$ etc.); likewise, elements of $(\mathrm{N}\,\mathcal{C})(x,y)$ are functors $\theta\colon \mathbf{2}^{x,y} \to \mathcal{C}$, i.e., correspond to commuting squares in $\mathcal{C}$ (writing 01 for $(x \mapsto 0, y \mapsto 1) \in \mathbf{2}^{x,y}$ etc.)

$$\begin{array}{ccc} \theta(01) & \xrightarrow{\theta(01\leq 11)} & \theta(11) \\ {\scriptstyle \theta(00\leq 01)}\uparrow & \nearrow{\scriptstyle \theta(00\leq 11)} & \uparrow{\scriptstyle \theta(10\leq 11)} \\ \theta(00) & \xrightarrow{\theta(00\leq 10)} & \theta(10) \end{array}$$

and so on for higher cubes. Note that $\mathrm{N}(\mathbf{2}^I)$ is in general not isomorphic to $\mathbf{y}I$ (for $I = x, y$, the cube in the nerve contains a diagonal which is not there in the standard cube).

Remark 2.8. Often, cubical sets are described as presheaves on a category dual to our cubical category; morphisms are given by certain $\mathbf{2}^J \to \mathbf{2}^I$ (namely those which come from $\mathbf{2}^f$ with $f\colon I \to J$ in the cubical category). This is used (for a variation of the cubical sets considered here) in [22, Section 4].

The following two definitions are crucial for the rest of this thesis.

Definition 2.9 (Non-dependent Path Space)**.** Let X be a cubical set. The (non-dependent) *path space* $[\mathbb{I}]X$ is defined by $([\mathbb{I}]X)(I) = X(I, x_I)$ (recall that x_I is a chosen fresh name for I). We can extend $f\colon I \to J$ to $(f, x_I = x_J)\colon I, x_I \to J, x_J$, and thus define the restriction along f of $\omega \in ([\mathbb{I}]X)(I)$ by $\omega f = \omega(f, x_I = x_J)$. (Note that on the left hand side the restriction is in $[\mathbb{I}]X$ whereas on the right hand side it is in X.) This defines a cubical set. Its points are $([\mathbb{I}]X)(\emptyset) = X(x_\emptyset)$, that is, the lines of X; its lines $([\mathbb{I}]X)(x) = X(x,y)$, y fresh, are the squares of X; and so on.

There is also an alternative definition of $[\mathbb{I}]X$: the elements $([\mathbb{I}]X)(I)$ are (equivalence classes) of the form $\langle x\rangle p$ where $x \notin I$ and $p \in X(I,x)$; two such elements $\langle x\rangle p$ and $\langle y\rangle q$ get identified if $p(x = y) = q \in X(I,y)$. For $f\colon I \to J$ we define $(\langle x\rangle p)f = \langle z\rangle(p(f, x = z))$ where z is some J-fresh name. The operation $\langle x\rangle-$ should be thought of as an abstraction- or binding operation. Correspondingly, there is also an application of $\langle x\rangle p \in ([\mathbb{I}]X)(I)$ to $a \in \{0,1\}$ or a a fresh name (i.e., $a \notin I$) given by

$$(\langle x\rangle p) \mathbin{@} a = p(x = a) \in X(I,a),$$

where $I, 0 = I, 1 = I$ by convention. This structure can be seen as the "affine" exponential of X by $\mathbb{I}$ (this can be made precise in the presentation of cubical sets as nominal sets, cf. 2.4).

Our first definition corresponds to choosing a canonical representative x_I for the bound name. We will mostly use this definition from now on but deviate whenever appropriate. But note that we also have the operation $\omega \mathbin{@} a =$

$\omega(x_I = a) \in X(I,a)$ for $a \in \{0,1\}$ or a fresh, with $\omega \in ([\mathbb{I}]X)(I)$, and the operation $\langle x \rangle p = p(x_I = x)$ for $p \in X(I,x)$.

Definition 2.10 (Non-dependent Identity Type)**.** For a cubical set X and two global sections $u, v \in X(\emptyset)$ we define the (non-dependent) *identity type* $\mathtt{Id}_X(u,v)$ to be the subobject $\mathtt{Id}_X(u,v) \subseteq [\mathbb{I}]X$ such that $\omega \in ([\mathbb{I}]X)(I)$ is an element of $(\mathtt{Id}_X(u,v))(I)$ if $\omega \mathbin{@} 0 = us_I$ and $\omega \mathbin{@} 1 = vs_I$, where $s_I \colon \emptyset \to I$ denotes the inclusion map; that is, ω is a line along x_I connecting u to v (more precisely, their degenerates).

More generally, we can also define $X \times X \vdash \mathtt{Id}_X$; for w_0 and w_1 in $X(I)$, $\mathtt{Id}_X(w_0, w_1)$ are those $\omega \in [\mathbb{I}]X$ such that $\omega \mathbin{@} 0 = w_0$ and $\omega \mathbin{@} 1 = w_1$. (The above cubical set $\mathtt{Id}_X(u,v)$ is given by substituting $X \times X \vdash \mathtt{Id}_X$ along $\langle u, v \rangle \colon 1 \to X \times X$.)

As an example, let us come back to the interval $\mathbb{I}$. We have two points $\vdash \mathtt{0} : \mathbb{I}$ and $\vdash \mathtt{1} : \mathbb{I}$ given by $\mathtt{0}_J = 0 \in \mathbb{I}(J)$ and likewise $\mathtt{1}_J = 1$. Those two points are equal via $\vdash \mathtt{seg} : \mathtt{Id}_{\mathbb{I}}(0,1)$ given by $\mathtt{seg}_J = \langle x \rangle x$ since $x \in \mathbb{I}(J,x)$ for x J-fresh.

Remark 2.11. It is important to note that this definition (or its dependent version we will see later) does *not* justify the axioms for an identity type in the CwF induced by **cSet**. Let us illustrate one of the requirements; assume we have a cubical set X and a dependent type over it, $X \vdash C$, and two global sections of X, $\vdash u, v : X$ (which correspond to two elements $u, v \in X(\emptyset)$). Moreover, we have an $\vdash \omega : \mathtt{Id}_X(u,v)$ and $\vdash p : C[u]$. The rules for the identity type require in particular an inhabitant of $\vdash C[v]$, i.e., we must be able to transport the element p along ω to an element in $C[v]$; but for arbitrary C there is no hope that we can achieve this as $C[v]$ might be empty even though $C[u]$ is not! Consider, e.g., the type $\mathbb{I} \vdash C$ defined for $\rho \in \mathbb{I}(I) = I \cup \mathbf{2}$ as $C\rho = \emptyset$ if $\rho \neq 0$, and $C\rho = \mathbb{N}$ if $\rho = 0$. The restriction maps $C\rho \to C\rho f$ is given by the unique map $\emptyset \to C\rho f$ if $\rho \neq 0$, and given by the identity on $\mathbb{N}$ if $\rho = 0$ (and thus also $\rho f = 0$). Now $\mathtt{seg} = \langle x \rangle x$ gives a term of type $\mathtt{Id}_{\mathbb{I}}(0,1)$, but there is no map from $C[0] = \mathbb{N}$ to $C[1] = \emptyset$. Hence we have to restrict the types in such a way that this property follows—this is the content of Chapter 3.

Remark 2.12. It is possible to justify the introduction rule for this identity types though: if $\vdash u : X$ given by the family $u_I \in X(I)$, define $\vdash \mathtt{refl}\, u : \mathtt{Id}_X(u,u)$ by $(\mathtt{refl}\, u)_I = u_I s_{x_I} \in X(I, x_I)$; note that $(\mathtt{refl}\, u)_I \in (\mathtt{Id}_X(u,u))(I)$ as

$$(\mathtt{refl}\, u)_I \mathbin{@} b = u_I.$$

Moreover, this defines a term as for $f \colon I \to J$,

$$(\mathtt{refl}\, u)_I f = (u s_{x_I}) f = u s_{x_I}(f, x_I = x_J) = (uf) s_{x_J}.$$

2.3 Cubical Sets via Nominal Sets

In this section we give equivalent categories to the category of cubical sets which are given by nominal sets with extra structure: so called 01-substitutions introduced in [36]. As studied in [38], the latter can also be presented as finitely supported M-sets for a suitable monoid M. The theory of nominal sets provides a mathematical theory of *names* based on symmetry. For more background on nominal sets we refer the reader to [37]. Here we follow [38].

We want to think of I-cubes u in a cubical set X, for say $I = x_1, \ldots, x_n$, as entities depending on the names $x_1, \ldots, x_n$, and can emphasize this by writing $u = u(x_1, \ldots, x_n)$ (similarly to indicate possible dependence of variables in a formula of predicate logic; see also Example 2.5). Now applying a morphism $f\colon I \to J$ should be thought of as applying a *substitution* of those names; the basic operations are renaming a variable into 0 or 1, renaming a variable into a fresh variable, and adding a (vacuous) variable dependency. E.g., if $n = 3$ and $f = (x_1 = y, x_2 = 0, x_3 = x_3)\colon I \to x_3, y, z$, we think of $uf \in X(x_3, y, z)$ as $u(y, 0, x_3)$ (with this notation, the application of a degeneracy map is not explicit).

Nominal sets capture this idea as well: each element in a nominal set depends on a *finite* set of names, and we can swap names (governed by suitable equations). The additional structure of 01-substitutions on a nominal set allows to set names to 0 or 1.

Definition 2.13. A *finite substitution* is a map $\pi\colon \mathbb{A} \to \mathbb{A} \cup \mathbf{2}$ such that $\{x \mid \pi x \neq x\}$ is finite. We denote the monoid of all finite substitutions by $\mathbf{Sb}$: its monoid operation is given by $\pi\pi'\colon \mathbb{A} \to \mathbb{A} \cup \mathbf{2}$ defined as $(\pi\pi')(x) = \pi'(\pi x)$ (note the order) if $\pi x \notin \mathbf{2}$ and $(\pi\pi')(x) = \pi x$ for $\pi x \in \mathbf{2}$; the unit 1 is given by the inclusion $\mathbb{A} \hookrightarrow \mathbb{A} \cup \mathbf{2}$. The element in $\mathbf{Sb}$ transposing x with y for $x, y \in \mathbb{A}$ is denoted by $(x\ y)$; the element $(x = b) \in \mathbf{Sb}$ for $x \in \mathbb{A}$ and $b \in \mathbf{2}$ sends x to b and is the identity otherwise.

Let us assume that $\mathbf{M}$ is a submonoid of $\mathbf{Sb}$.

Definition 2.14. The category of $\mathbf{M}$-*sets* is simply the presheaf category $[\mathbf{M}, \mathbf{Set}]$ where $\mathbf{M}$ is considered as a category (with one element). So such an $\mathbf{M}$-set Γ is given by a set Γ and an action $\Gamma \times \mathbf{M} \to \Gamma$ satisfying

$$\rho\, 1 = \rho \qquad (\rho\, \pi)\, \pi' = \rho\, (\pi\pi')$$

for $\rho \in \Gamma$ and $\pi, \pi' \in \mathbf{M}$.

A finite subset $I \subseteq \mathbb{A}$ *supports* an element $\rho \in \Gamma$ if for all $\pi, \pi' \in \mathbf{M}$,

$$\forall x \in I (\pi x = \pi' x) \to \rho\, \pi = \rho\, \pi'. \tag{2.2}$$

The *category of finitely supported* $\mathbf{M}$-*sets* $[\mathbf{M}, \mathbf{Set}]_{\mathsf{fs}}$ is the full subcategory of $\mathbf{M}$-sets for whose objects Γ every $\rho \in \Gamma$ is finitely supported (i.e., has a support which is finite).

The submonoid **Cb** of **Sb** contains those $\pi \in \mathbf{Sb}$ satisfying for $\pi x, \pi y \notin \mathbf{2}$:

$$\pi x = \pi y \to x = y$$

This condition is like the condition for morphisms in the cubical category $\square$. This entails the following lemma.

Lemma 2.15. *For all* $f\colon I \to J$ *in* $\square$ *there exists a* $\pi \in \mathbf{Cb}$ *for which* $\pi x = fx$ *for all* $x \in I$.

Proof. See [38]. $\square$

Theorem 2.16. *The category of finitely supported* **Cb***-sets* $[\mathbf{Cb}, \mathbf{Set}]_{\mathsf{fs}}$ *is equivalent to the category of cubical sets* **cSet**.

Proof. We only define the functor $F\colon \mathbf{cSet} \to [\mathbf{Cb}, \mathbf{Set}]_{\mathsf{fs}}$ and its inverse G on objects; for the detailed proof we refer the reader to [38]. For a cubical set X we take the colimit of X in $\square$ restricted to inclusions $I \subseteq J$, and define $FX = \varinjlim_I X(I)$. So an element in FX is an equivalence class $[I, u]$ with I a finite set of atoms and $u \in X(I)$; two such equivalence classes $[I, u]$ and $[J, v]$ are equal iff for some $K \supseteq I \cup J$, $u\iota = v\iota'$ where ι and ι' are the inclusions $I \subseteq K$ and $J \subseteq K$, respectively. For $\pi \in \mathbf{Cb}$ we have that $\pi|_I\colon I \to \pi(I) - \mathbf{2}$ is a morphism in $\square$, and we define $[I, u]\,\pi = [\pi(I) - \mathbf{2}, u(\pi|_I)]$.

Conversely, given a finitely supported **Cb**-set Y, GY is defined by

$$(GY)(I) = \{u \in Y \mid u \text{ is supported by } I\}.$$

For $u \in (GY)(I)$ and $f\colon I \to J$, $uf := u\,\pi$ with π as in Lemma 2.15; this doesn't depend on the choice of π since I supports u, and is well defined since $u\pi$ is supported by $\pi(I) - \mathbf{2} \subseteq J$ (cf. [38, Corollary 2.5]). $\square$

Let Γ be a finitely supported **Cb**-set and $u \in \Gamma$; then u there is a least set of names supporting u denoted by $\mathrm{supp}(u)$ (cf. [38, Definition 2.6]). Then a set of atoms I supports u iff $\mathrm{supp}(u) \subseteq I$. We write $u \# v$ for if $\mathrm{supp}(u) \cap \mathrm{supp}(v) = \emptyset$ for $u, v \in \Gamma$. The set $\mathbb{A} \cup \mathbf{2}$ becomes a finitely supported **Cb**-set via $x\pi = \pi(x)$, $0\pi = 0$, and $1\pi = 1$, whose corresponding cubical set is isomorphic to the interval $\mathbb{I}$; clearly $\mathrm{supp}(x) = \{x\}$ and $\mathrm{supp}(0) = \mathrm{supp}(1) = \emptyset$.

Another way to present finitely supported **Cb**-sets is as nominal sets with extra structure.

Definition 2.17. The group $\mathrm{Per}(\mathbb{A})$ is given by the permutations in **Sb**, i.e., bijections $\pi\colon \mathbb{A} \to \mathbb{A}$ such that $\{x \in \mathbb{A} \mid \pi x \neq x\}$ is finite. The category of *nominal sets* **Nom** is the category of finitely supported $\mathrm{Per}(\mathbb{A})$-sets.

Each finitely supported **Cb**-set is also a nominal set. The notion of support w.r.t. the **Cb**-set structure coincides with the notion of support of nominal sets.

Definition 2.18. Let Γ be a nominal set. A structure of 01-*substitutions* on Γ is given by operations $(\!(x = c)\!)\colon \Gamma \to \Gamma$ for each name x and $c \in \mathbf{2}$ satisfying for $u \in \Gamma$:

(a) $(u((x = c)))\pi = u\pi((\pi x = c))$;

(b) $u((x = c)) \# x$;

(c) $u \# x \to u((x = c)) = u$;

(d) $x \neq y \to u((x = c))((y = d)) = u((y = d))((x = c))$.

The nominal sets with 01-substitutions constitute the object of the category **01Nom** whose morphisms are morphisms $\sigma \colon \Gamma \to \Delta$ in **Nom** such that $(\sigma(u))((x = c)) = \sigma(u((x = c)))$.

Lemma 2.19. *The categories* **01Nom** *and finitely supported* **Cb***-sets are equivalent. And thus the former is also equivalent to* **cSet***.*

Proof. See [38]. The main idea is to use that any element in **Cb** is a composition of swaps $(x\ y)$ and $(x = a)$. The latter are taken care of by the structure of 01-substitutions $((x = a))$. ☐

2.4 Separated Products

The equivalence of cubical sets with nominal sets equipped with 01-substitutions lets us translate important constructions on nominal sets to cubical sets. Let X and Y be cubical sets and $u \in X(I)$; recall that we wrote $x \# u$ if u degenerate along $x \in I$. If also $v \in Y(I)$, write $u \# v$ if there are $u' \in X(J)$ and $v' \in X(K)$ for $J, K \subseteq I$ with $J \cap K = \emptyset$ such that $u = u'\iota$ and $v = v'\iota'$ with ι and ι' being the respective inclusions $J \subseteq I$ and $K \subseteq I$. In that case $uf \# vf$ for $f \colon I \to I'$ witnessed by $u'(f|J)$ and $v'(f|K)$ and $f(J) \cap f(K) \subseteq \mathbf{2}$ since f is injective on names.

The *separated product* $X * Y$ of X and Y is given by

$$(X * Y)(I) = \{(u, v) \in X(I) \times Y(I) \mid u \# v\} \subseteq (X \times Y)(I)$$

and componentwise restrictions, making it a sub-cubical set of $X \times Y$.

It is easily verified that for J, K disjoint, we have

$$\mathbf{y}J * \mathbf{y}K \cong \mathbf{y}(J \cup K).$$

Moreover, $- * Y$ extends to a functor which has a right adjoint $Y \multimap -$, which we have already seen in Definition 2.9 as $[\mathbb{I}]-$ in the special case of $Y = \mathbb{I}$.

The functor on cubical sets $Y \multimap -$ is given by

$$(Y \multimap Z)(I) = \mathbf{cSet}(Y * \mathbf{y}I, Z)$$

for Z in **cSet** (this is natural in I and Z, and thus induces a endo-functor on **cSet**). That this is adjoint to $- * Y$ follows from the fact that $- * Y$ commutes with colimits.

Let us sketch that $[\mathbb{I}]X$ is isomorphic to $\mathbb{I} \multimap X$. Define the map $\varphi \colon [\mathbb{I}]X \to (\mathbb{I} \multimap X)$ as follows: let $(a, f) \in (\mathbb{I} * \mathbf{y}I)(J)$, i.e., $a \in \mathbb{I}(J)$ and $f \colon I \to J$ with

$a \# f$; the latter yields that $f = f'\iota$ with $f' \colon I \to J - a$ and ι the inclusion $I - a \subseteq I$. We define for $\langle x \rangle \omega$ in $([\mathbb{I}]X)(J)$

$$\varphi(\langle x \rangle \omega)(a, f) = \omega(f', x = a) \in X(J).$$

One can check that $\varphi(\langle x \rangle \omega)$ is an element in $(\mathbb{I} \multimap X)(J)$, and that φ is a morphism. The inverse $\psi \colon (\mathbb{I} \multimap X) \to [\mathbb{I}]X$ is given for $\theta \in (\mathbb{I} \multimap X)(I)$ by

$$\psi\theta = \langle x \rangle \theta(x, s_x)$$

for x fresh, so $x \# s_x$. One can check that this defines a morphism, which is indeed an inverse of φ.

Chapter 3

Kan Cubical Sets

As we have seen in the last chapter it is not possible to justify the elimination rules for the identity types defined from path spaces in the cubical set model. In this chapter we will introduce the notion of when a type $\Gamma \vdash A$ is a *uniform Kan type*; restricting to types with this condition is sufficient to justify the elimination rule for identity types defined from path spaces. The main work is to show that this notion is closed under the type formers.

3.1 The Uniform Kan Condition

The uniform Kan condition is about requiring fillers of "open box" like shapes. It is reminiscent of Daniel Kan's original extension axiom in [28]. Kan introduced this notion in order to give a combinatorial definition of homotopy groups. To simplify the discussion of the filling condition we will first only introduce the non-relative case. Let us first introduce these open box shapes and operations on them; these shapes correspond to *horns* in simplicial sets.

Definition 3.1 (Open Boxes)**.** Let I be a finite sets of names, $x, J \subseteq I$ with $x \notin J$ and $a \in \{0, 1\}$. The triple $S = ((x, a); J; I)$ is called an *open box shape* on I; its *indices* $\langle S \rangle$ are given by

$$\langle S \rangle = \{(y, c) \mid c \in \mathbf{2}, y \in J, x \text{ and } (y, c) \neq (x, a)\} = \{(x, \bar{a})\} \cup J \times \mathbf{2}.$$

If $a = 1$, we call S a $+$-shape; otherwise, i.e., $a = 0$, a $-$-shape.

Let X be a cubical set. An S-*open box* in X (or simply *open box*) is given by a family $\vec{u}$ indexed by $\langle S \rangle$ such that $u_{yb} \in X(I - y)$ for $(y, b) \in \langle S \rangle$ and such that $\vec{u}$ is *adjacent compatible*, i.e., for all $(y, b), (z, c) \in \langle S \rangle$ with $y \neq z$ we have

$$u_{yb}(z = c) = u_{zc}(y = b) \tag{3.1}$$

The element $u_{x\bar{a}}$ of an S-open box $\vec{u}$ is called the *principal side* of $\vec{u}$ and x its *principal direction*; the sides u_{yc}, for $y \in J$, are called its *non-principal sides*, and J its *non-principal directions*. We assume that the first entry v in $\vec{u} = v, \vec{v}$ is its principal side.

For $f\colon I \to K$ with $x, J \subseteq \operatorname{def}(f)$ and an S-open box $\vec{u}$ in I, we define the open box $\vec{u}f$ given by the components $u_{yb}(f - y) \in X(K - fy)$; this gives in an Sf-open box where $Sf = ((fx, a); fJ; K)$ for $S = ((x, a); J; I)$ and where fJ denotes the image of J under f.

Definition 3.2 (Kan Cubical Set)**.** A *uniform Kan cubical set* X, or simply *Kan cubical set*, is a cubical set X equipped with the following *filling operations*. For each open box shape S and S-open box $\vec{u}$ in X, we require and element

$$[X]_S \vec{u} \in X(I)$$

such that for $(y, b) \in \langle S \rangle$

$$([X]_S \vec{u})(y = b) = u_{yb}$$

and additionally for each $f\colon I \to K$ with $x, J \subseteq \operatorname{def}(f)$ the following *uniformity condition* (or *coherence condition*):

$$([X]_S \vec{u})f = [X]_{Sf}(\vec{u}f) \tag{3.2}$$

If S is a $+$-shape, we denote $[X]_S \vec{u}$ by $X{\uparrow}_S \vec{u}$; and if S is a $-$-shape by $X{\downarrow}_S \vec{u}$. Moreover, we usually suppress the shape of the box and tacitly assume that an open box fits the filling operator. We also give names to the face in the principal direction

$$X^+ \vec{u} = (X{\uparrow}\vec{u})(x = 1) \qquad \text{and} \qquad X^- \vec{u} = (X{\downarrow}\vec{u})(x = 0)$$

and call them the *induced composition operations*.

It is also possible to consider a cubical set X with only *composition operations*: for each open box shape $S = ((x, a); J; I)$ and S-open box we require $|X|_S \vec{u} \in X(I - x)$ such that for $(y, b) \in \langle S \rangle$, $(|X|_S \vec{u})(y = b) = u_{yb}(x = a)$ and for $f\colon I - x \to J$ defined on $I - x$, we require the uniformity conditions

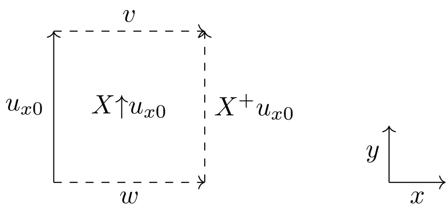
$$(|X|_S \vec{u})f = |X|_{Sf'}(\vec{u}f')$$

with $f' = (f, x = z)\colon I \to J, z$ for z fresh w.r.t. J. (One should consider the name x in $|X|_S \vec{u} \in X$ to be *bound*.) The induced composition operations of a Kan cubical set are composition operations in this sense.

We emphasize that the above definition requires a *fixed choice* of fillers of open boxes and is thus equipped with “algebraic” operations, and this algebraic presentation is crucial in order to formulate the uniformity condition (3.2). A similar notion for simplicial sets is that of an algebraic Kan complex [34]. But note that these come without additional equations like the uniformity condition above. Similar operations for semi-simplicial sets have been considered in [4].

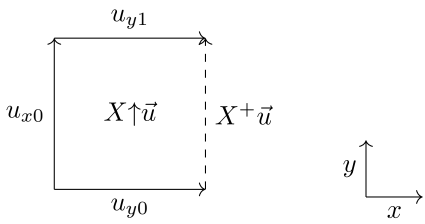
We will sometimes refer to the operations of a Kan cubical set as its *Kan structure*.

Let us analyze the filling operations of a uniform Kan cubical set X in low dimensions. The simplest case is where J is empty (with the same notations as in the definition above): a corresponding open box, for say $a = 1$, is simply given by an element $u_{x0} \in X(I - x)$; the filler $X{\uparrow}u_{x0}$ is an element in $X(I)$ with $(X{\uparrow}u_{x0})(x = 0) = u_{x0}$. Thus if say $y \in I$, this gives a square:

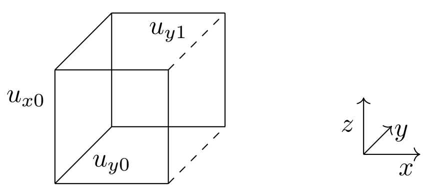


By definition the right hand face is the composition $X^{+}u_{x0} \in X(I - x)$. But what about the top and bottom faces v and w? The uniformity conditions guarantee that the filling operation commutes with the face operations $(y = 1)$ and $(y = 0)$, and thus we know that $v = X{\uparrow}(u_{x0}(y = 1))$ and $u = X{\uparrow}(u_{x0}(y = 0))$. Moreover, if u_{x0} happens to be degenerate along y, the uniformity condition entails that $X{\uparrow}u_{x0} = vs_y$.

Let us now assume that $J = y$; then an open box of the corresponding shape comes with three elements: $\vec{u} = u_{x0}, u_{y0}, u_{y1}$ such that $u_{x0} \in X(I - x)$ and $u_{y0}, u_{y1} \in X(I - y)$, and that $u_{x0}(y = b) = u_{yb}(x = 0)$ for $b \in \mathbf{2}$. Thus the situation can be depicted as a "U-shape" whose filler $X{\uparrow}\vec{u} \in X(I)$ is as indicated:



If we also have another $z \in I$ this situation also can be depicted as follows, where the filler $X{\uparrow}\vec{u}$ is the whole cube (omitting the arrow tips):



The uniformity conditions ensure that the top face $(z = 1)$ of the cube is the filler of the top "U-shape" $\vec{u}(z = 1)$; and likewise for the bottom. And similarly, if *all* of the sides in $\vec{u}$ are degenerate in direction z, the filling cube is the degenerate of the filling square of the top "U-shape" (which is in this case equal to the lower one).

And so on: with J having two elements, corresponding open boxes become cubes with a missing side and missing interior etc.

(But note that the uniformity conditions say nothing about how two filling operations for different cardinalities of the J parameter relate.)

Lemma 3.3. *Let R be a commutative ring. The cubical set $P = R[\cdot]$ induced by R as defined in Example 2.5 is a Kan cubical set.*

Proof. Let $S = ((x,a);J;I)$ and $\vec{p}$ an open box of shape S in P. We will construct the filler $p = [P]_S\vec{p}$ by an iterated linear interpolation. First we define $p_J \in R[I]$ such that $p_J(y=b) = p_{yb}$ for $y \in J$ by induction on the size of J: we start with $p_\emptyset = 0$; if $J = z, K$, we set

$$p_{z,K} = p_K + (1-z)(p_{z0} - p_K(z=0)) + z(p_{z1} - p_K(z=1)).$$

Note that if $K = \emptyset$, this p_z is simply the linear interpolation

$$p_z = (1-z)p_{z0} + zp_{z1}.$$

We have for $b \in \mathbf{2}$, $p_{z,K}(z=b) = p_K(z=b) + p_{zb} - p_K(z=b) = p_{zb}$ and for $y \in K$ with the IH and the fact that $\vec{p}$ is adjacent compatible:

$$\begin{aligned} p_{z,K}(y=b) &= p_K(y=b) + (1-z)\big(p_{z0}(y=b) - p_K(y=b)(z=0)\big) \\ &\quad + z\big(p_{z1}(y=b) - p_K(y=b)(z=1)\big) \\ &= p_{yb} + (1-z)\big(p_{yb}(z=0) - p_{yb}(z=0)\big) \\ &\quad + z(p_{yb}(z=1) - p_{yb}(z=1)) = p_{yb} \end{aligned}$$

As a last step we define the filler $p \in R[I]$ in case $a = 1$ by

$$p = p_J + (1-x)(p_{x0} - p_J(x=0))$$

and analogously for $a = 0$. This has the correct faces as is readily checked. The definition satisfies the uniformity conditions: observe that $(p_J)f$ for f defined on J is the same as the corresponding polynomial for the $p_{yb}(f-y)$ with $y \in J$; similarly, this extends to pf for f defined on x, J. □

Lemma 3.4. *The cubical nerve $\mathrm{N}(\mathcal{G})$ of a (small) groupoid $\mathcal{G}$ is a Kan cubical set.*

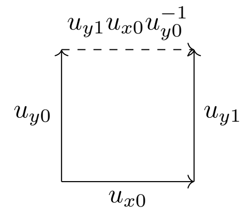
Proof. Let $S = ((x,1);J;I)$ and $\vec{u}$ an S-open box in $\mathrm{N}(\mathcal{G})$. The proof for $-$-open boxes is similar. In case $J = \emptyset$, we define the filling of $\vec{u}$ (which only consists of u_{x0}) by us_x. This clearly satisfies the required uniformity conditions. In case J contains at least two elements, we argue that the input box $\vec{u}$ already contains all the needed edges and can be uniquely considered as an I-cube: we define the filler $u\colon \mathbf{2}^I \to G$ on an object $\alpha\colon I \to \mathbf{2}$ by $u\alpha = u_{yb}(\alpha - y)$ for $\alpha y = b$ with $(y,b) \in \langle S \rangle$. Note that there has to exist such a (y,b) since $J \neq \emptyset$. Also, this is well defined since $\vec{u}$ is adjacent compatible. On morphisms $\alpha \leq \beta$ we first define $u(\alpha \leq \beta)$ as follows. If both $\alpha y = \beta y = b$

for some $(y, b) \in \langle S \rangle$, then we take $u_{yb}(\alpha - y \leq \beta - y)$. Otherwise, there are $(y, b), (z, c) \in \langle S \rangle$ with $\alpha y = b$ and $\beta z = c$ with $y \neq z$ since J contains at least two elements; then we take

$$u_{zc}((\beta - z, x = 0) \leq \beta - z) \circ u_{x0}(\alpha - x \leq \beta - x) \circ (u_{yb}((\alpha - y, x = 0) \leq \alpha - y))^{-1}$$

which is forced by the groupoid structure. This determines u uniquely from the fact that u has to have $\vec{u}$ as corresponding faces.

It remains the case where J consists exactly of one element, say y. We construct the filler by induction on $I - (x, J)$ together with showing that the filler is unique, and hence satisfies the required uniformity conditions. In case $I - (x, J)$ is empty, we construct the composition as follows:



This also determines the filler $u \in (\mathrm{N}(\mathcal{G}))(x, y)$ since the diagram commutes, and is unique by the groupoid structure. Now in case, $I - (x, J)$ contains z, we inductively fill $\vec{u}(z = 0)$ and $\vec{u}(z = 1)$ to unique u_{z0} and u_{z1} in $(\mathrm{N}(\mathcal{G}))(I - z)$, respectively. Now we define the filler $u \in (\mathrm{N}(\mathcal{G}))(I)$ of $\vec{u}$ as the filler of the extended open box $\vec{u}, u_{z0}, u_{z1}$ (which we already constructed above). If $u' \in (\mathrm{N}(\mathcal{G}))(I)$ is another filler of $\vec{u}$, then by IH, $u'(z = b)$ has to be equal to u_{zb}, and hence to $u(z = b)$. But then u and u' are both also fillers of $\vec{u}, u_{z0}, u_{z1}$ which we have shown to be unique above. □

Definition 3.5. Let Γ be a cubical set. A type $\Gamma \vdash A$ is a *uniform Kan type*, or simply *Kan type*, if it is equipped with the following operations. Let $\alpha \in \Gamma(I)$, S an open box shape on I, and $\vec{u}$ an S-open box in $A\alpha$, i.e., $\vec{u}$ is an $\langle S \rangle$-indexed family where the component u_{yb} for $(y, b) \in \langle S \rangle$ is an element $u_{yb} \in A\alpha(y = b)$, such that $\vec{u}$ is adjacent compatible. We require fillers

$$[A\alpha]_S \vec{u} \in A\alpha$$

such that for $(y, b) \in \langle S \rangle$

$$([A\alpha]_S \vec{u})(z = b) = u_{yb}$$

and additionally for each $f : I \to K$ with $x, J \subseteq \mathrm{def}(f)$ the *uniformity condition* holds, i.e.,

$$([A\alpha]_S \vec{u})f = [A\alpha]_{Sf}(\vec{u}f).$$

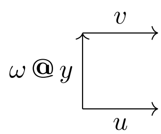
We will use the analogous notation $A\alpha{\uparrow}\vec{u}, A\alpha{\downarrow}\vec{u}, A\alpha^{+}\vec{u}, A\alpha^{-}\vec{u}$ as in Definition 3.2.

Remark 3.6. To get the definition of when a map of cubical sets $\sigma\colon \Delta \to \Gamma$ is a (uniform) Kan fibration replace $A\alpha$ by $\sigma^{-1}(\alpha)$ in the above definition. Then $\Gamma \vdash A$ is a uniform Kan type iff the projection $\mathtt{p}\colon \Gamma.A \to \Gamma$ is a uniform Kan fibration.

As an immediate consequence of the definition we get:

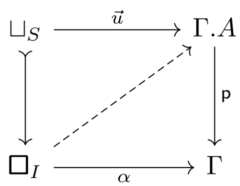
Lemma 3.7. *A X cubical set is a Kan cubical set if and only if, X considered as a type in the empty context $1 \vdash X$ is a Kan type.*

Remark 3.8. The open box shapes with non-principal faces appear naturally when we want that the operation $u \mapsto A\rho{\uparrow}u$ extends to the identity type, say, $X \times X \vdash \mathtt{Id}_X$ for a cubical set X. Given $u, v \in X(I, x)$ and $\omega \in \mathtt{Id}_X(u_0, v_0)$ (where $u_0 = u(x = 0)$ etc.), so $\omega\,@\,0 = u_0$ and $\omega\,@\,1 = v_0$; let y be fresh. A filler $\mathtt{Id}_X(u, v){\uparrow}\omega$ is a filler of the following open box shape (modulo the abstraction $\langle y\rangle -$):

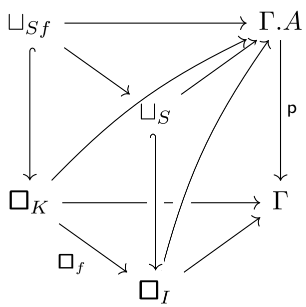


Thus it is natural to require these more general filling operations on X.

Remark 3.9. Similar to the lifting condition for simplicial sets w.r.t. horn inclusions, we can formulate the uniform Kan condition of a type $\Gamma \vdash A$ as the lifting of maps. Let $\sqcup_S(K) \subseteq \square_I(K)$ for $S = ((x, a); J; I)$ consist of those $f\colon I \to K$ such that $fy = b$ for some $(y, b) \in \langle S\rangle$. This defines a sub-cubical set of $\square_I$. Open boxes of shape S in a cubical set Γ correspond to morphisms $\sqcup_S \to \Gamma$. The existence of fillers are *chosen* diagonal fillers for every outer square (where the lower horizontal map corresponds to an I-cube in Γ, and the upper horizontal map to an open box):



For $f\colon I \to K$ defined on all of x, J the map (given by the Yoneda embedding) $\square_f : \square_K \to \square_I$ restricts to a map $\sqcup_{Sf} \to \sqcup_S$ (since for $g \in \sqcup_{Sf}(L)$, i.e., $gz = b$ for $(z, b) \in \langle Sf\rangle$, we have $z = fy$ for some $y \in J$, and thus $fg = \square_f g \in \sqcup_S(L)$). The uniformity conditions translate to the commutation of the following prism where the two diagonal (slightly bent) arrows into $\Gamma.A$ are the chosen fillers:



Remark 3.10. Note that for $S = ((x,a);J;I)$ with $I = x,J,K$, where x,J and K disjoint, we have

$$\sqcup_S \cong \sqcup_{((x,a);J;J)} * \square_K. \tag{3.3}$$

Moreover, for $f\colon I \to I'$ defined on x,J, we can write $I' = fx, fJ, K'$ (disjoint). We have the induced map

$$\sqcup_{((x,a);J;J)} * \square_K \cong \sqcup_S \longrightarrow \sqcup_{Sf} \cong \sqcup_{((fx,a);fJ;fJ)} * \square_{K'}$$

whose right component is induced by the renaming on x,J and whose left component is induced by the morphism $f - x,J\colon K \to K'$.

Now this suggests yet another reformulation of the Kan structure on a cubical set due to Peter Lumsdaine of which we only give a short sketch. Let $\sqcup^a_{x;J} := \sqcup_{((x,a);J;J)} \subseteq \square_{x,J}$ and consider the cubical set $\sqcup^a_{x;J} \multimap X$ for a cubical set X: by the Yoneda Lemma, its K-cubes are

$$\sqcup^a_{x;J} * \square_K \to X$$

which in case that x,J and K are disjoint corresponds to (by (3.3))

$$\sqcup_{((x,a);J;x,J,K)} \to X,$$

i.e., the $((x,a);J;x,J,K)$ open boxes in X. Consider the canonical map

$$r^a_{x;J}\colon (\square_{x,J} \multimap X) \to (\sqcup^a_{x;J} \multimap X)$$

induced by the inclusion $\sqcup^a_{x;J} \subseteq \square_{x,J}$.

Assume now that X has a Kan structure. We define a section $s^a_{x;J}$ of $r^a_{x;J}$. This amounts to give level-wise sections

$$(\sqcup^a_{x;J} \multimap X)(K) \to (\square_{x,J} \multimap X)(K)$$

natural in K. But the left hand side corresponds to open boxes $\vec{u}$ of shape $((x,a);J;x,J,K)$, and the right hand side corresponds to x,J,K cubes, and thus we can use $[X]_S\vec{u}$ to define the image of $\vec{u}$. Now this is natural in K since we have the uniformity conditions for maps $f\colon x,J,K \to x,J,K'$ which are the identity on x, J. The uniformity conditions for renaming in x, J yield additional equations on the sections: a renaming $f\colon x, J \to fx, fJ$ defined on all of x, J induces the vertical maps in

$$\begin{array}{ccc} (\square_{x,J} \multimap X) & \rightleftarrows & (\sqcup^a_{x;J} \multimap X) \\ \downarrow & & \downarrow \\ (\square_{fx,fJ} \multimap X) & \rightleftarrows & (\sqcup^a_{fx;fJ} \multimap X) \end{array} \tag{3.4}$$

where the horizontal maps are given by the corresponding sections and retractions. Now the (full) uniformity conditions yield that the diagram commutes. The converse is also true: if we have a choice of sections for all the $r^a_{x;J}$ such that all the squares of the form (3.4) commute, then X has a Kan structure.

3.2 The Kan Cubical Set Model

In this section we show that Kan types are closed under the type formers and gives rise to a CwF supporting Σ-, and Π-types. It is crucial to observe that the filling operations are part of the definition when $\Gamma \vdash A$ is a Kan type. That means that for Kan types $\Gamma \vdash A$ and $\Gamma \vdash B$ we can have $A = B$ as cubical sets but *not* necessarily as Kan types, i.e., their Kan structures might not coincide. Thus we have to check *coherence conditions*, i.e., we have to verify that the CwF equations between types are preserved by the filling operations.

Theorem 3.11. *Kan types give rise to a CwF supporting* Π- *and* Σ-*types, where*

1. *contexts* $\Gamma \vdash$ *are interpreted by cubical sets;*
2. *substitutions* $\sigma\colon \Delta \to \Gamma$ *are maps of cubical sets;*
3. *types* $\Gamma \vdash A$ *are Kan types;*
4. *terms of a Kan type* $\Gamma \vdash A$ *are terms of* $\Gamma \vdash A$ *considered as a type in the cubical set (i.e., presheaf) sense.*

Note that by Section 1.2 and the fact that cubical sets are just presheaves on the cubical category $\square$, we get that cubical sets induce a CwF. The difference to the model we give in the theorem is in item 3, that is, types are equipped with a Kan structure. The proof of Theorem 3.11 spans the rest of this section and the definition of the respective Kan structures are given in the proofs of the following theorems. The part of the CwF which is shared with the CwF of cubical sets is given in the same manner. So context extension and the definition of the required terms and context morphisms are defined to be the same as for the cubical set model. Also, the constructions on types are the same except that we have to care about the Kan structure as well. In other words, forgetting the Kan structure induces a morphism of CwFs preserving Π and Σ.

Let us also mention (without proof) that the model also supports base types like the natural numbers. Their interpretation is given via the constant presheaf.

We will reserve "type" for a type in the cubical set sense and use "Kan type" for a type with its Kan structure.

Theorem 3.12. *If $\Gamma \vdash A$ is a Kan type, $\Delta \vdash$, and $\sigma\colon \Delta \to \Gamma$ a context morphism, then also the type $\Delta \vdash A\sigma$ is a Kan type. Moreover, the Kan structure is such that we get:*

$$A\mathbf{1} = A \qquad (A\sigma)\tau = A(\sigma\tau)$$

Proof. For an I-cube α of Δ recall that we defined $(A\sigma)\alpha = A(\sigma\alpha)$. We define the filling operations of $(A\sigma)\alpha$ to be those of $A(\sigma\alpha)$, i.e., we set $[(A\sigma)\alpha]_S\vec{u} = [A(\sigma\alpha)]_S\vec{u}$ for an open box $\vec{u}$. With this definition it is clear that A and $A\mathbf{1}$ have the same Kan structure, and likewise for the other equation. □

Theorem 3.13. *If $\Gamma \vdash A$ and $\Gamma.A \vdash B$ are Kan types, then so is the type $\Gamma \vdash \Sigma AB$. Moreover, $(\Sigma AB)\sigma = \Sigma(A\sigma)(B(\sigma\mathsf{p}, \mathsf{q}))$ as Kan types.*

Proof. Let $\vec{u}$ be an S-open box in $(\Sigma AB)\alpha$ for $\alpha \in \Gamma(I)$. Then with $w_{yb} = (u_{yb}, v_{yb})$ where $u_{yb} \in A\alpha(y = b)$ and $v_{yb} \in B(\alpha(y = b), u_{yb})$ for $(y, b) \in \langle S\rangle$, we get that $\vec{u}$ is an S-open box in $A\alpha$ which we can fill to $u = [A\alpha]_S\vec{u}$. Now $\vec{v}$ is also a S-open box in $B(\alpha, u)$ and we set

$$[(\Sigma AB)\alpha]_S\vec{w} = (u, [B(\alpha, u)]_S\vec{v}).$$

This definitions satisfies the required uniformity conditions as they are satisfied for $\Gamma \vdash A$ and $\Gamma.A \vdash B$.

Now if $\alpha = \sigma\beta$ for $\sigma\colon \Delta \to \Gamma$, we get that $u = [(A\sigma)\beta]_S\vec{u}$ and

$$[B(\sigma\beta, u)]_S\vec{v} = [(B(\sigma\mathsf{p}, \mathsf{q}))(\beta, u)]_S\vec{v},$$

yielding $(\Sigma AB)\sigma = \Sigma(A\sigma)(B(\sigma\mathsf{p}, \mathsf{q}))$ as Kan types. □

For the next theorem we need some notations. Given $\Gamma \vdash A$, $\alpha \in \Gamma(I)$, $u \in A\alpha$, and a shape $S = ((x, a); J; I)$, we define an S-open box u^S by "carving" out an S-shape from u, i.e., u^S is given by the components $u^S_{yc} = u(y = c) \in A\alpha(y = c)$. Note that the filling operation $[A\alpha]_S$ is a section of this operation $(-)^S$. Moreover, for $f\colon I \to J$ defined on J, x we have $(u^S)f = u^{Sf}$.

For $\Gamma \vdash \Pi AB$ and S-shapes $\vec{w}$ in $(\Pi AB)\alpha$ and $\vec{u}$ in $A\alpha$ such that $u^S = \vec{u}$ for some $u \in A\alpha$, we define the S-shape $\vec{w}\vec{u}$ in $B(\alpha, u)$ by the components $(\vec{w}\vec{u})_{yc} = (w_{yc})_{\mathbf{1}} u_{yc} \in B(\alpha(y = c), u(y = c))$. (Recall from Section 1.2.1 that elements in Π-types are *families* of dependent functions.) For $f\colon I \to J$ defined on J, x this satisfies $(\vec{w}\vec{u})f = (\vec{w}f)(\vec{v}f)$ since:

$$\begin{aligned}(\vec{w}\vec{u})_{yc}(f - y) &= ((w_{yc})_{\mathbf{1}} u_{yc})(f - y) \\ &= (w_{yc})_{(f-y)}(u_{yc}(f - y)) \\ &= (w_{yc}(f - y))_{\mathbf{1}}(u_{yc}(f - y))\end{aligned}$$

Theorem 3.14. *If* $\Gamma \vdash A$ *and* $\Gamma.A \vdash B$ *are Kan types, then so is* $\Gamma \vdash \Pi AB$. *Moreover,* $(\Pi AB)\sigma = \Pi(A\sigma)(B(\sigma\mathsf{p}, \mathsf{q}))$ *as Kan types.*

Proof. Let $C = \Pi AB$ and let S be an open box shape. We assume that S is a $+$-shape, i.e., of the form $S = ((x, 1); J; I)$; the case of $-$-shapes is symmetric.

First, we will define the composition operations $C\alpha^+_S \vec{w} \in (\Pi AB)\alpha(x = 1)$ for $\alpha \in \Gamma(I)$ and $\vec{w}$ an S-open box in $(\Pi AB)\alpha$. This amounts to define a family of dependent functions $(C\alpha^+_S \vec{w})_f$ in $\prod_{u \in A\alpha(x=1)f} B(\alpha(x = 1)f, u)$ for all $f\colon I - x \to K$, such that

$$\big((C\alpha^+_S \vec{w})_f(u)\big)g = (C\alpha^+_S \vec{w})_{fg}(ug). \tag{3.5}$$

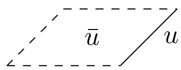
We will first define $(C\alpha^+_S w)_f$ for $f = \mathbf{1}\colon I - x \to I - x$. For this let $u \in A\alpha(x = 1)$. We use the Kan fillings with shape $S_x = ((x, 0); \emptyset; I)$ to extend u to $A\alpha{\downarrow_x} u \in A\alpha$ (with $\downarrow_x$ for $\downarrow_{S_x}$), of which we carve out an S-shape we then apply to $\vec{w}$, and map the result up:

$$(C\alpha^+_S \vec{w})_{\mathbf{1}}(u) = B(\alpha, A\alpha{\downarrow_x} u)^+_S(\vec{w}\,(A\alpha{\downarrow_x} u)^S) \tag{3.6}$$

which is in $B(\alpha(x = 1), u)$ as $(A\alpha{\downarrow_x} u)(x = 1) = u$. This defines $(C\alpha^+_S \vec{w})_{\mathbf{1}}$ for arbitrary α and w. Note that to give the open box $(A\alpha{\downarrow_x} u)^S$ we only need composition operations of $\Gamma \vdash A$ (not so in the argument to B).

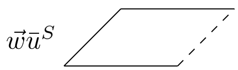
Let us illustrate this (where we assume one non-principal direction y). We are given $\vec{w}$

and we are given u which we fill to $\bar{u} = A\alpha{\downarrow_x} u$



in $A\alpha$

of which we carve out the open box $\bar{u}^S$ and apply $\vec{w}$, which we then fill in $B(\alpha, \bar{u})$



(3.7)

where the last dashed line on the right is the definition of $(C\alpha^+_S \vec{w})_{\mathbf{1}}$.

For general $f\colon I - x \to K$ we define $(C\alpha^+_S \vec{w})_f$ as follows. In case there is $(y, c) \in \langle S\rangle$ such that $fy = c$ we write f as $f = (y = c)(f - y)$ with $(f - y)\colon I - x, y \to K$. (Note that $y \neq x$.) We define

$$(C\alpha^+_S \vec{w})_f = (w_{yc})_{(x=1)(f-y)}. \tag{3.8}$$

This is well defined since $\vec{w}$ is adjacent compatible. Note that this ensures

$$(C\alpha^+_S \vec{w})(y = c) = w_{yc}(x = 1).$$

Otherwise, i.e., f is defined on J we let z be fresh w.r.t. K (e.g., take $z = x_K$) and set

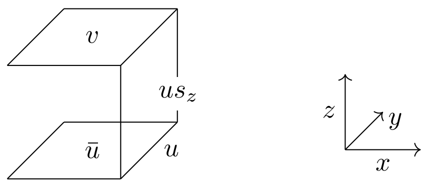
$$(C\alpha^+_S\vec{w})_f = \big((C\alpha f^z_x)^+_{Sf^z_x}(\vec{w}f^z_x)\big)_{\mathbf{1}} \tag{3.9}$$

where f^z_x is $(f, x = z)\colon I \to K, z$. By the uniformity conditions, this definition does not depend on the choice of z, and we also get by uniformity, (3.6), and the discussion above

$$\big((C\alpha^+_S\vec{w})_{\mathbf{1}}(u)\big)f = \big((C\alpha f^z_x)^+_{Sf^z_x}(\vec{w}f^z_x)\big)_{\mathbf{1}}(uf). \tag{3.10}$$

Note, that (3.9) says that the family $(C\alpha^+_S\vec{w})_f$ is determined by the value at $f = \mathbf{1}$ (with different α, S and $\vec{w}$). The uniformity condition follows from (3.9) (note that the left hand side is $((C\alpha^+_S\vec{w})f)_{\mathbf{1}}$ by definition). Equation (3.5) follows from (3.10) together with (3.8) and (3.9); more formally, to prove (3.5) one distinguishes cases on the definedness of f. If f is not defined on one of the non-principal sides, (3.5) follows from (3.8). Otherwise, the left hand side in (3.5) is given by (3.9), in which case one distinguishes cases on the definedness of g: in case g is not defined on one of the corresponding non-principal sides one uses (3.8) again, and otherwise, uses (3.10). Thus we obtain an element in $C\alpha(x = 1)$.

Next we define $C\alpha{\uparrow_S}\vec{w} \in C\alpha$; we do so again by first defining $(C\alpha{\uparrow_S}\vec{w})_f$ for $f = \mathbf{1}\colon I \to I$. Let $v \in A\alpha$, $u = v(x = 1)$, and let z be fresh (e.g., $z = x_I$). Consider the shape $S_{x,z} = ((x, 0); z; I)$; we get an $S_{x,z}$ open box us_z, $A\alpha{\downarrow_x}u$, v in $A\alpha s_z$ (where us_z is the principal side and the next two sides are at $(z, 0)$ and $(z, 1)$ respectively), illustrated as (again only with one non-principal side y, $\bar{u} = A\alpha{\downarrow_x}u$, and all sides should be considered "solid")



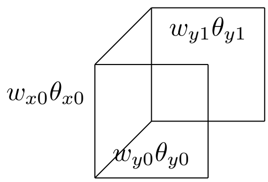
which we fill to

$$\theta = A\alpha s_z{\downarrow_{x,z}}(us_z, A\alpha{\downarrow_x}u, v) \in A\alpha s_z$$

where we wrote ${\downarrow_{x,z}}$ for ${\downarrow_{S_{x,z}}}$. From this we carve out an $Ss_z = ((x, 1); J; I, z)$ open box and apply it to $\vec{w}s_z$ to get an Ss_z open box

$$(\vec{w}s_z)\,\theta^{Ss_z} \text{ in } B(\alpha s_z, \theta), \tag{3.11}$$

or as picture (where we write $w_{x0}\theta_{x0}$ for $(w_{x0})_{s_z}(\theta(x = 0))$ etc.)



Notice that if we take the face $(z = 0)$ in the above picture, we get the same open box $\vec{w}\vec{u}^S$ as in (3.7).

We define the open box χ in $B(\alpha s_z, \theta)$ of shape $((z,1); x, J; I, z)$ with the principal side

$$B(\alpha, A\alpha{\downarrow_S}u){\uparrow_S}(\vec{w}\,(A\alpha{\downarrow_x}u)^S)$$

(which is the square in (3.7)) and where the non-principal side at $(x,1)$ is $\big((C\alpha^+_S\vec{w})_{\mathbf{1}}(u)\big)s_z$; these are compatible by construction (3.6); the non-principal side at $(x,0)$ is given by the principal side of the open box (3.11); the non-principal sides in directions J are the non-principal sides of (3.11).

We fill this to obtain the definition

$$(C\alpha{\uparrow}\vec{w})_{\mathbf{1}}(v) = B(\alpha s_z, \theta)^+\chi \tag{3.12}$$

yielding an element in $B(\alpha, v)$ since $\theta(z = 1) = v$ by definition of θ. This concludes the definition of $((C\alpha){\uparrow}\vec{w})_{\mathbf{1}}$ for all α and $\vec{w}$.

For general $f\colon I \to K$ we define $(C\alpha{\uparrow}\vec{w})_f$ by distinguishing cases:

$$(C\alpha{\uparrow_S}\vec{w})_f = \begin{cases} (C\alpha^+_S\vec{w})_{(f-x)} & \text{if } fx = 1,\\ (w_{yc})_{(f-y)} & \text{if } fy = c \text{ for some } (y,c) \in \langle S\rangle,\\ (C\alpha f{\uparrow_{Sfx}}\vec{w}f)_{\mathbf{1}} & \text{otherwise, i.e., } f \text{ is defined on } x, J. \end{cases}$$

where $(f - x)\colon I - x \to K$ and $(f - y)\colon I - y \to K$.

To conclude that this definition is a well-defined element of $C\alpha$ and satisfies the uniformity condition we need to verify that

$$\big((C\alpha{\uparrow}\vec{w})_{\mathbf{1}}(v)\big)f = (C\alpha{\uparrow}\vec{w})_f(vf) \tag{3.13}$$

for $f\colon I \to K$. If for some $y \in x, J$, fy is not defined, (3.13) follows from how the open box χ is defined. Otherwise, i.e., f is defined on all x, J, (3.13) follows by inspecting that, in the definition of $(C\alpha{\uparrow}\vec{w})_{\mathbf{1}}(v)$,

$$(B(\alpha s_z, \theta)^+\chi)f = B(\alpha f s_{z'}, \theta f')^+(\chi f')$$

where $f' = (f, z = z')$ and z' fresh w.r.t. K. And $\theta f'$ and $\chi f'$ are, by uniformity, exactly those arguments appearing in the definition of $(C\alpha f{\uparrow}\vec{w}f)_{\mathbf{1}}(vf)$ which is the right hand side of (3.13).

To verify that the Kan structure of $\Pi(A\sigma)(B(\sigma\mathsf{p},\mathsf{q}))$ (as defined above) is equal to the Kan structure for $(\Pi\, A\, B)\sigma$ (as defined in the proof of Theorem 3.12), assume that above $\alpha = \sigma\beta$ for $\sigma\colon \Delta \to \Gamma$; then $C\alpha = ((\Pi\, A\, B)\sigma)\beta$ and in equation (3.6) we have

$$\begin{aligned} &B(\sigma\beta, A(\sigma\beta){\downarrow_x}u)^+(\vec{w}(A(\sigma\beta){\downarrow_x}u)^S)\\ &= (B(\sigma\mathsf{p},\mathsf{q}))(\beta, (A\sigma)\beta{\downarrow_x}u)^+(\vec{w}((A\sigma)\beta{\downarrow_x}u)^S) \end{aligned}$$

and the right hand side is the definition of $\big(\Pi(A\sigma)(B(\sigma\mathsf{p},\mathsf{q}))^+\vec{w}\big)_1(u)$. Similarly for the other parts of the definition. □

Remark 3.15. We also get a CwF if in Theorem 3.11, we require contexts to be *Kan* cubical sets instead of just cubical sets. The crucial point that this works is that if $\Gamma \vdash$ is a Kan cubical set and $\Gamma \vdash A$ is a Kan type, then $\Gamma.A \vdash$ is a Kan cubical set. The proof of this statement is along the lines of the closure of Kan types under Σ-types (see Theorem 3.13); in fact, it can be derived using that $1 \vdash \Gamma$ is a Kan type.

3.3 Identity Types

We will now define the identity type of a cubical set and justify an elimination operator and functional extensionality for it. The underlying idea of the definition of identity type is that a proof of equality $\mathtt{Id}_A(a, b)$ should be a path with endpoints a and b.

Recall the definition of the non-dependent path space (Definition 2.9), non-dependent identity types (Definition 2.10), and the notation used there.

Definition 3.16. Let $\Gamma \vdash A$, $\Gamma \vdash a : A$, $\Gamma \vdash b : A$. The *identity type* $\Gamma \vdash \mathtt{Id}_A(a, b)$ is defined as follows: for $\rho \in \Gamma(I)$ an element $\omega \in (\mathtt{Id}_A(a, b))\rho$ is an element $\omega \in A(\rho s_{x_I})$ such that $\omega(x_I = 0) = a\rho$ and $\omega(x_I = 1) = b\rho$. The restriction by a $f \colon I \to J$ of $\omega \in (\mathtt{Id}_A(a, b))\rho$ is defined, as for the non-dependent identity type, by $\omega f = \omega(f, x_I = x_J)$ (where on right we use the restriction of A).

As in 2.10 we could have used equivalence classes $\langle x\rangle p$ with $p \in A(\rho s_x)$ where $\rho \in \Gamma(I)$, x fresh for I, and $\langle x\rangle p = \langle y\rangle q$ if $p(x = y) = q$. The operation $\omega \mathbin{@} a$ for $a \in \mathbf{2}$ or a fresh is defined as there, i.e., $\omega \mathbin{@} a = \omega(x_I = a)$. For $p \in A\rho s_x$ we set $\langle x\rangle p = p(x_I = x)$. Note that for $f \colon I \to J$ and $\omega \in (\mathtt{Id}_A(a, b))\rho$, $x \notin I$ we have $(\omega \mathbin{@} x)(f, x = a) = \omega f \mathbin{@} a$. No matter which definition is used we have the notions of $\langle x\rangle p$ and $\omega \mathbin{@} a$ as in 2.10.

Theorem 3.17. *Kan types are closed under identity types, i.e., if* $\Gamma \vdash A$ *is a Kan type,* $\Gamma \vdash a : A$, *and* $\Gamma \vdash b : A$, *then* $\Gamma \vdash \mathtt{Id}_A(a, b)$ *is a Kan type. Moreover, for* $\sigma \colon \Delta \to \Gamma$, $(\mathtt{Id}_A(a, b))\sigma = \mathtt{Id}_{A\sigma}(a\sigma, b\sigma)$ *as Kan types.*

Proof. Let $\vec{\omega}$ be an S-open box in $(\mathtt{Id}_A(a, b))\rho$ with $\rho \in \Gamma(I)$ and let z be fresh; then $\vec{\omega} \mathbin{@} z$ (component-wise) is an S-open box in $A\rho s_z$. We extend this to an S, z-open box in $A\rho s_z$, where S, z is like S but with the non-principal side z added, given by $\vec{\omega} \mathbin{@} z, a\rho, b\rho$ and define

$$[\mathtt{Id}_A(a, b)\rho]_S \vec{\omega} = \langle z\rangle\, [A\rho s_z]_{S,z}(\vec{\omega} \mathbin{@} z, a\rho, b\rho) \tag{3.14}$$

Note that by the definition of the extended open box

$$([\mathtt{Id}_A(a, b)\rho]_S \vec{\omega}) \mathbin{@} 0 = a\rho \text{ and } ([\mathtt{Id}_A(a, b)\rho]_S \vec{\omega}) \mathbin{@} 1 = b\rho.$$

The uniformity conditions follow from those in A.

For a substitution $\sigma \colon \Delta \to \Gamma$ an element ω of $(\mathtt{Id}_A(a, b)\sigma)\rho = \mathtt{Id}_A(a, b)(\sigma\rho)$ is given by $\omega \mathbin{@} z$ in $A((\sigma\rho)s_z) = (A\sigma)(\rho s_z)$ with $\omega \mathbin{@} 0 = a(\sigma\rho) = (a\sigma)\rho$ and $\omega \mathbin{@} 1 = b(\sigma\rho) = (b\sigma)\rho$. Hence $\mathtt{Id}_A(a, b)\sigma = \mathtt{Id}_{A\sigma}(a\sigma, b\sigma)$ as types, and similarly this holds for the Kan structure. $\square$

Note that for the filling operations in $\mathtt{Id}_A(a,b)$ we need the filling operations in A with *one more non-principal direction.* This is the main reason to require operations with non-principal sides!

Lemma 3.18. *For $\Gamma \vdash A$ and $\Gamma \vdash a : A$ we have $\Gamma \vdash \mathtt{refl}\, a : \mathtt{Id}_A(a,a)$, and $(\mathtt{refl}(a))\sigma = \mathtt{refl}(a\sigma)$ for a substitution $\sigma \colon \Delta \to \Gamma$.*

Proof. For $\rho \in \Gamma(I)$ define $(\mathtt{refl}\, a)\rho = a(\rho s_{x_I})$, i.e., $(\mathtt{refl}\, a)\rho = \langle x \rangle a(\rho s_x)$. This defines a term as $(a(\rho s_{x_I}))f = a(\rho s_{x_I}(f, x_I = x_J)) = a((\rho f)s_{x_J}) = (\mathtt{refl}\, a)(\rho f)$ for $f \colon I \to J$. □

Note that the previous lemma does not rely on $\Gamma \vdash A$ to be a Kan type.

Next, we will define an elimination operator for the identity type. We will define various operations which together define the J-eliminator where the usual definitional equality holds only up to propositional equality (i.e., we will give an inhabitant of the respective Id-type).

First we define the *transport* along a path. Let $\Gamma \vdash A$ be a type and $\Gamma.A \vdash C$ be a Kan type. Furthermore let $\Gamma \vdash a : A$, $\Gamma \vdash b : A$, $\Gamma \vdash e : C[a]$, and $\Gamma \vdash d : \mathtt{Id}_A(a,b)$. (Recall that $[a]$ is the substitution $(1,a) \colon \Gamma \to \Gamma.A$.) We define a term $\Gamma \vdash \mathtt{subst}_C(d,e) : C[b]$ as follows. For $\rho \in \Gamma(I)$ and a fresh $x = x_I$ we have that $d\rho \mathbin{@} x \in A\rho s_x$ with $(d\rho \mathbin{@} x)(x=0) = a\rho$ and $(d\rho \mathbin{@} x)(x=1) = b\rho$. Thus $(\rho s_x, d\rho \mathbin{@} x)$ connects $[a]\rho$ to $[b]\rho$ along x. We define

$$\mathtt{subst}_C(d,e)\rho = C(\rho s_x, d\rho \mathbin{@} x)^+_x(e\rho) \in C[b]\rho \tag{3.15}$$

where the composition operation is w.r.t. the shape $((x,1);\emptyset;I)$. This defines a term, since by the uniformity conditions we get for $f \colon I \to J$ and $y = x_J$ J-fresh:

$$\begin{aligned}
(\mathtt{subst}_C(d,e)\rho)f &= \big(C(\rho s_x, d\rho \mathbin{@} x)^+_x(e\rho)\big)f \\
&= \big(C(\rho s_x, d\rho \mathbin{@} x)(f, x=y)\big)^+_y(e\rho f) \\
&= C(\rho f s_y, d\rho f \mathbin{@} y)^+_y(e\rho f) \\
&= \mathtt{subst}(d,e)(\rho f)
\end{aligned}$$

where we used that $s_x(f, x=y) = f s_y$.

If $\sigma \colon \Delta \to \Gamma$, then $(\mathtt{subst}_C(d,e))\sigma = \mathtt{subst}_{C(\sigma\mathtt{p},\mathtt{q})}(d\sigma, e\sigma)$ which is readily checked from the defining equation (3.15).

According to the definition of $\mathtt{subst}$ the line $C(\rho s_x, d\rho \mathbin{@} x)\!\uparrow_x\!(e\rho)$ connects $e\rho$ to $\mathtt{subst}_C(d,e)\rho$. In particular, for $d = \mathtt{refl}\, a$ we get

$$\Gamma \vdash \mathtt{substEq}_C(a,e) : \mathtt{Id}_{C[a]}(e, \mathtt{subst}_C(\mathtt{refl}\, a, e))$$

where

$$\mathtt{substEq}_C(a,e)\rho = \langle x \rangle\, C(\rho s_x, a\rho s_x)\!\uparrow_x\!(e\rho). \tag{3.16}$$

Similar to above one can show that this defines a term and is stable under substitution.

Definition 3.19. For a type $\Gamma \vdash A$ we define the type $\Gamma \vdash \mathtt{contr}(A)$ by

$$\mathtt{contr}(A) = \Sigma A\,(\Pi A\mathsf{p}(\mathtt{Id}_{A\mathsf{pp}}(\mathsf{qp}, \mathsf{q})))$$

or using “informal” type theoretical notation

$$\mathtt{contr}(A) = (\Sigma x : A)\,(\Pi y : A)\ \mathtt{Id}_A(x, y).$$

We say that a type $\Gamma \vdash A$ is *contractible* if the type $\Gamma \vdash \mathtt{contr}(A)$ is inhabited. In this case we call the first projection a center of contraction.

Next, we show that if $\Gamma \vdash A$ is a Kan type and $\Gamma \vdash a : A$, then the *singleton type* $\mathtt{singl}(A, a) = \Sigma A\,\mathtt{Id}_{A\mathsf{p}}(a\mathsf{p}, \mathsf{q})$ is contractible. We clearly have $\Gamma \vdash (a, \mathtt{refl}\,a) : \mathtt{singl}(A, a)$. We now show that $(a, \mathtt{refl}\,a)$ is also a center of contraction, i.e., we have to give a term

$$\Gamma.\,\mathtt{singl}(A, a) \vdash \mathtt{isCenter}(a) : \mathtt{Id}((a, \mathtt{refl}\,a)\mathsf{p}, \mathsf{q}).$$

where we omitted the subscript $\mathtt{singl}(A, a)\mathsf{p}$. Let $\rho \in \Gamma(I)$ and $(b, \omega) \in \mathtt{singl}(A, a)\rho$, so $b \in A\rho$ and $\omega \in (\mathtt{Id}_{A\mathsf{p}}(a\mathsf{p}, \mathsf{q}))(\rho, b)$. For a fresh $x = x_I$, $\omega \mathbin{@} x \in A\rho s_x$ connects $a\rho$ to b. Let y be a fresh name ($y = x_{I,x}$). We have to give an element in $\mathtt{Id}((a, \mathtt{refl}\,a)\mathsf{p}, \mathsf{q})(\rho, (b, \omega))$ for which we give an element in $\mathtt{singl}(A, a)\rho s_y$ connecting $(a, \mathtt{refl}\,a)\rho$ to (b, ω). This amounts to give a pair whose first component α is in $A\rho s_y$ and connects a to b and whose second component in $(\mathtt{Id}_{A\mathsf{p}}(a\mathsf{p}, \mathsf{q}))(\rho s_y, \alpha)$ connects $(\mathtt{refl}\,a)\rho$ to ω along y. Consider the open box $(a\rho s_y, a\rho s_x, \omega \mathbin{@} x)$ in $A(\rho s_x s_y)$:

$$\begin{array}{ccc} a\rho & & b \\ {\scriptstyle a\rho s_x}\uparrow & & \uparrow{\scriptstyle \omega\,@\,x} \\ a\rho & \xrightarrow{a\rho s_y} & a\rho \end{array} \tag{3.17}$$

Its filler gives rise to the second component and its composition, i.e., its upper face gives the first component. Thus we define

$$\begin{aligned} \mathtt{isCenter}(a)(\rho, (b, \omega)) = \langle y \rangle \big(& A(\rho s_x s_y)^{+}_{x,y}(a\rho s_y, a\rho s_x, \omega \mathbin{@} x), \\ & \langle x \rangle\ A(\rho s_x s_y){\uparrow}_{x,y}(a\rho s_y, a\rho s_x, \omega \mathbin{@} x)\big) \end{aligned} \tag{3.18}$$

The uniformity conditions guarantee that this defines a section.

Let us define the more common elimination operator of Paulin-Mohring from the above operations—with the difference that the usual definitional equality is only propositional. To not make the notation too heavy we'll use informal reasoning in type theory; note that the definition can be given internally in type theory and we don't refer to the model; this definition follows Danielsson's Agda development[1] accompanying [15]. First note that using the transport operation `subst` one can define composition $p \centerdot q : \mathtt{Id}_A(a, c)$ of two identity proofs $p : \mathtt{Id}_A(a, b)$, $q : \mathtt{Id}_A(b, c)$, as well as inverses $p^{-1} : \mathtt{Id}_A(b, a)$.

[1] Available at `www.cse.chalmers.se/~nad/`.

Let A be a type, $a : A$, and $C(b,p)$ a type given $b : A$, $p : \mathtt{Id}_A(a,b)$, such that $v : C(a, \mathtt{refl}\,a)$; for $b : A$ and $p : \mathtt{Id}_A(a,b)$ we define $\mathtt{J}(a,v,b,p) : C(b,u)$. We can consider C as a dependent type over $\mathtt{singl}(A,a)$ via $C(\mathsf{p}w, \mathsf{q}w)$ for $w : \mathtt{singl}(A,a)$. As we showed in the last paragraph, $\mathtt{singl}(A,a)$ is contractible with center $(a, \mathtt{refl}\,a)$, and thus we get a witness $\mathtt{app}(\varphi, (b,p)) : \mathtt{Id}((a, \mathtt{refl}\,a), (b,p))$ for $\varphi = \lambda\, \mathtt{isCenter}(a)$; now with $\mathtt{subst}$ (w.r.t. the type $C(\mathsf{p}w, \mathsf{q}w)$ for $w : \mathtt{singl}(A,a)$) we can define

$$\mathtt{J}(a,v,b,p) = \mathtt{subst}((\mathtt{app}(\varphi, (a, \mathtt{refl}\,a)))^{-1} \centerdot \mathtt{app}(\varphi, (b,p)), v). \tag{3.19}$$

Note that we now are able to derive $\mathtt{Id}_{\mathtt{Id}_A(a,a)}(p^{-1} \centerdot p, \mathtt{refl}\,a)$ for all $p : \mathtt{Id}_A(a,b)$ using $\mathtt{J}$ and $\mathtt{substEq}$.

It remains to check the propositional equality for $\mathtt{J}$. If, in (3.19), $p = \mathtt{refl}\,a$ and $b = a$, we get that

$$\mathtt{app}(\varphi, (a, \mathtt{refl}\,a))^{-1} \centerdot \mathtt{app}(\varphi, (b,p))$$

is propositionally equal to $\mathtt{refl}(\mathtt{refl}\,a)$, and thus using $\mathtt{subst}$ and $\mathtt{substEq}$ again one gets a witness of $\mathtt{Id}_{C(a,\mathtt{refl}\,a)}(v, \mathtt{J}(a,v,a,\mathtt{refl}\,a))$. This concludes the sketch that $\mathtt{J}$ with the rewrite rule as propositional equality is definable from $\mathtt{subst}, \mathtt{substEq}$, and $\mathtt{isCenter}$—and that alone in type theory without referring to their actual semantics.

Note that to get the propositional equality for $\mathtt{J}$ we could not use $\mathtt{subst}$ on $\mathtt{app}(\varphi, (b,p))$ directly.

3.3.1 Functional Extensionality

The equality on the function space is extensional.

Theorem 3.20. *Equality on* Π*-types is extensional, i.e., any pointwise equal functions are equal. More precisely, given* $\Gamma \vdash A$, $\Gamma.A \vdash B$ *we can justify the rule:*

$$\frac{\Gamma \vdash w' : \Pi A B \qquad \begin{array}{c}\Gamma \vdash w : \Pi A B \\ \Gamma \vdash e : \Pi A\ \mathtt{Id}_{B[q]}(\mathtt{app}(w\mathsf{p}, \mathsf{q}), \mathtt{app}(w'\mathsf{p}, \mathsf{q}))\end{array}}{\Gamma \vdash \mathtt{funExt}(w, w', e) : \mathtt{Id}(w, w')}$$

Proof. Let $\rho \in \Gamma(I)$, $x = x_I$ fresh, and write θ for $\mathtt{funExt}(w, w', e)$. We have to define a dependent function

$$(\theta\rho)_f \text{ in } \prod_{u \in A(\rho s_x f)} B(\rho s_x f, u) \text{ for each } f\colon I, x \to J.$$

In case $fx = 0$ we set $(\theta\rho)_f = w\rho_{(f-x)}$, and likewise, in case $fx = 1$ we set $(\theta\rho)_f = w'\rho_{(f-x)}$. For f defined on x we set $(\theta\rho)_f = (\theta(\rho f))_{\mathbf{1}}$ so that we can assume $f = \mathbf{1}\colon I, x \to I, x$. This definition ensures $(\theta\rho)g = \theta(\rho g)$. Let $u \in A\rho s_x$ and $u_b = u(x = b) \in A\rho$; we have to define $(\theta\rho)_{\mathbf{1}} u \in B(\rho s_x, u)$

connecting $w\rho_{\mathbf{1}}u_0$ to $w'\rho_{\mathbf{1}}u_1$ along x. Let y be fresh. We get that $(e\rho_{\mathbf{1}}u_1) \mathbin{@} y \in B(\rho s_y, u_1)$ and hence we define $(\theta\rho)_{\mathbf{1}}u$ to be the filler of the following box

$$\begin{array}{ccc} w\rho_{\mathbf{1}}u_0 & \xrightarrow{(\theta\rho)_{\mathbf{1}}u} & w'\rho_{\mathbf{1}}u_1 \\ {\scriptstyle (w\rho_{\mathbf{1}}u_0)s_y}\uparrow & & \uparrow{\scriptstyle (e\rho_{\mathbf{1}}u_1) \mathbin{@} y} \\ w\rho_{\mathbf{1}}u_0 & \xrightarrow[w\rho_{s_x}u]{} & w\rho_{\mathbf{1}}u_1 \end{array} \quad \text{over } B(\rho s_x s_y, u s_y).$$

That is, $(\theta\rho)_{\mathbf{1}}u = B(\rho s_x s_y, u s_y)^{+}_{y,x}(w\rho_{s_x}u, (w\rho_{\mathbf{1}}u_0)s_y, (e\rho_{\mathbf{1}}u_1) \mathbin{@} y)$. Now if $f\colon I, x \to J$ is defined on x and z J-fresh, we get by the uniformity conditions

$$\begin{aligned} ((\theta\rho)_{\mathbf{1}}u)f &= \big(B(\rho s_x s_y, u s_y)^{+}_{y,x}(w\rho_{s_x}u, (w\rho_{\mathbf{1}}u_0)s_y, (e\rho_{\mathbf{1}}u_1) \mathbin{@} y)\big)f \\ &= \big(B(\rho s_x s_y, u s_y)(f, y = z)\big)^{+}_{z,fx} \\ &\qquad \big((w\rho_{s_x}u)f, (w\rho_{\mathbf{1}}u_0)(f - x)s_z, (e\rho_{\mathbf{1}}u_1)(f - x) \mathbin{@} z\big) \end{aligned}$$

which is the same as $(\theta\rho)_f(uf)$. In case f is not defined on x we get $((\theta\rho)_{\mathbf{1}}u)f = (\theta\rho)_f(uf)$ by the above definition. We leave the verification of $(\theta\rho)f = \theta(\rho f)$ from these equations to the reader. □

3.3.2 Path Application

Although in general `subst` and `J` don't satisfy the usual definitional equalities the model justifies another operation `ap` which satisfies new definitional equalities which don't hold if we define the operation using `J`.

Theorem 3.21. *Let $\Gamma \vdash A$ and $\Gamma \vdash B$ be Kan types, $\Gamma \vdash u : A$, and $\Gamma \vdash v : A$. Then we can validate the rule*

$$\frac{\Gamma \vdash \varphi : A \to B \qquad \Gamma \vdash w : \mathtt{Id}_A(u, v)}{\Gamma \vdash \mathtt{ap}(\varphi, w) : \mathtt{Id}_B(\mathtt{app}(\varphi, u), \mathtt{app}(\varphi, v))}$$

satisfying

$$\begin{aligned} \mathtt{ap}(\mathrm{id}, w) &= w \\ \mathtt{ap}(\varphi \circ \psi, w) &= \mathtt{ap}\big(\varphi, \mathtt{ap}(\psi, w)\big) \\ \mathtt{ap}(\varphi, \mathtt{refl}\, a) &= \mathtt{refl}(\mathtt{app}(\varphi, a)) \\ \mathtt{ap}(\lambda(b\mathtt{p}), w) &= \mathtt{refl}\, b \end{aligned}$$

where id *is the identity function, $\circ$ denotes composition, and $\lambda(b\mathtt{p})$ is the constant b function. Moreover this operation is stable under substitution, i.e.,* $\mathtt{ap}(\varphi, w)\sigma = \mathtt{ap}(\varphi\sigma, w\sigma)$ *for $\sigma\colon \Delta \to \Gamma$.*

Proof. For $\rho \in \Gamma(I)$ and a fresh x we set

$$\mathtt{ap}(\varphi, w)\rho = \langle x\rangle\, (\varphi\rho)_{s_x}(w\rho \mathbin{@} x).$$

This defines a term as for $f\colon I \to J$ and y J-fresh we get

$$\begin{aligned}(\mathtt{ap}(\varphi, w)\rho)f &= \langle y\rangle\, ((\varphi\rho)_{s_x}(w\rho \mathbin{@} x))(f, x = y)\\ &= \langle y\rangle\, (\varphi\rho)_{s_x(f,x=y)}(w\rho \mathbin{@} x(f, x = y))\\ &= \langle y\rangle\, (\varphi\rho)_{f s_y}(w(\rho f) \mathbin{@} y)\\ &= \langle y\rangle\, (\varphi(\rho f))_{s_y}(w(\rho f) \mathbin{@} y) = \mathtt{ap}(\varphi, w)(\rho f)\end{aligned}$$

The other equations immediately follow from the definition. Let us, for example, check the second equation: $\varphi \circ \psi$ is $\lambda\, \mathtt{app}(\varphi \mathtt{p}, \mathtt{app}(\psi \mathtt{p}, \mathtt{q}))$ and hence

$$\begin{aligned}\mathtt{ap}(\varphi \circ \psi, w)\rho \mathbin{@} x &= ((\varphi \circ \psi)\rho)_{s_x}(w\rho \mathbin{@} x)\\ &= \mathtt{app}(\varphi \mathtt{p}, \mathtt{app}(\psi \mathtt{p}, \mathtt{q}))(\rho s_x, w\rho \mathbin{@} x)\end{aligned}$$

which using $((\varphi \mathtt{p})(\rho s_x, w\rho \mathbin{@} x))_{\mathbf{1}} = (\varphi\rho)_{s_x}$ (and analogously for ψ) becomes

$$\begin{aligned}&= (\varphi\rho)_{s_x}((\psi\rho)_{s_x}(w\rho \mathbin{@} x))\\ &= (\varphi\rho)_{s_x}(\mathtt{ap}(\psi, w)\rho \mathbin{@} x)\\ &= (\mathtt{ap}(\varphi, \mathtt{ap}(\psi, w)))\rho \mathbin{@} x.\end{aligned}$$

□

3.3.3 Heterogeneous Identity Types

The model also comes with a natural notion of *heterogeneous identity types* satisfying the following rules

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash u_0 : A \qquad \Gamma \vdash u_1 : A \qquad \Gamma \vdash p : \mathtt{Id}_A(u_0, u_1) \\ \Gamma.A \vdash C \qquad \Gamma \vdash v_0 : C[u_0] \qquad \Gamma \vdash v_1 : C[u_1]}{\Gamma \vdash \mathtt{HId}^p_C(v_0, v_1)}$$

$$\mathtt{HId}^{\mathtt{refl}\, u}_C(v_0, v_1) = \mathtt{Id}_{C[u]}(v_0, v_1)$$

omitting the equations for stability under substitution. Its interpretation in the cubical set model is given by: for $\rho \in \Gamma(I)$ the set $(\mathtt{HId}^p_C(v_0, v_1))\rho$ contains elements $\langle x\rangle w$ with $w \in C(\rho s_x, p\rho \mathbin{@} x)$ (up to renaming of bound variables, or making a canonical choice $x = x_I$ as for the identity type) such that $w(x = b) = v_b$ for $b \in \mathbf{2}$. The Kan structure is given by the equation:

$$[(\mathtt{HId}^p_C(v_0, v_1))\rho]_S \vec{\omega} = \langle z\rangle\, [C(\rho s_z, p\rho \mathbin{@} z)]_{S,z}(\vec{\omega}, v_0\rho, v_1\rho).$$

The accompanying equation is immediate given the definition of Id.

3.4 Regular Kan Types

We have seen that in general the usual definitional equality for J holds only up to propositional equality. If we restrict to Kan types which satisfy a regularity condition we can give a definition of J for which the equality is definitional. This notion is however *not* preserved by all the type formers.

Definition 3.22. A Kan type $\Gamma \vdash A$ is *regular* if for any open box shape S with principal direction x and S-open box $\vec{u}$ in ρs_x such that each component u_{yb} with $y \neq x$ is degenerated along x, i.e., $u_{yb} = v_{yb} s_x$ for some $v_{yb} \in A\rho(y = b)$, then the filling satisfies

$$[A\rho s_x]_S \vec{u} = u_{xa} s_x$$

where u_{xa} is the principal face of $\vec{u}$. (So $\vec{u} = u_{xa}, \vec{v} s_x$.)

Theorem 3.23. *Regularity is preserved under substitution and the type formers* Id *and* Σ*, that is:*

1. *If* $\Gamma \vdash A$ *is a regular Kan type, then so is* $\Delta \vdash A\sigma$ *for* $\sigma \colon \Delta \to \Gamma$*.*
2. *If* $\Gamma \vdash A$ *be a regular Kan type,* $\Gamma \vdash a : A$*, and* $\Gamma \vdash b : A$*, then also* $\Gamma \vdash \mathtt{Id}_A(a, b)$ *is regular.*
3. *If* $\Gamma \vdash A$ *and* $\Gamma.A \vdash B$ *are regular Kan types, then so is* $\Gamma \vdash \Sigma AB$*.*

Proof. The proof is by analyzing the definitions of the fillings. For (1) recall that the fillings in $(A\sigma)\rho s_x$ are defined by fillings in $A(\sigma(\rho s_x))$. Since σ commutes with degenerates, regularity is preserved to $A\sigma$.

For (2) the defining equation (3.14) yields for an S-open box $\omega, \vec{\omega}$ in $\mathtt{Id}_A(a, b)\rho s_x$ degenerate along x (where x is the principal direction of S and ω the principal side of the box)

$$\begin{aligned}[\mathtt{Id}_A(a, b)\rho s_x]_S(\omega, \vec{\omega}) &= \langle z \rangle\, [A\rho s_x s_z]_{S,z}(\omega \mathbin{@} z, \vec{\omega} \mathbin{@} z, a\rho s_x, b\rho s_x) \\ &= \langle z \rangle\, (\omega \mathbin{@} z) s_x = \omega s_x\end{aligned}$$

as $s_x s_z = s_z s_x$ and $(\omega \mathbin{@} z, \vec{\omega} \mathbin{@} z, a\rho s_x, b\rho s_x)$ is also degenerate along x.

For (3) one readily checks from the defining equations in Theorem 3.13. □

Let us sketch how one can use regularity in order to define a variant of J given in Section 3.3 but with the right definitional equality for regular Kan types (using the notations from Section 3.3): First, the definition of `subst` and `isCenter` is as in Section 3.3. Note that by the regularity condition (for C) $\mathtt{substEq}_C(a, e)$ is simply $\mathtt{refl}(e)$ in equation (3.16) on page 42. Moreover, the definition of $\mathtt{isCenter}(a)$ on page 43 is such that if $b = a\rho$ and $\omega = a\rho s_x$ the square (3.17) is degenerate. Second, this can be used to directly define a variant J′ of J by

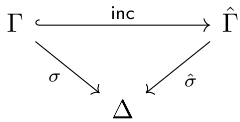
$$\mathtt{J}'(a, v, b, p) = \mathtt{subst}(\mathtt{app}(\varphi, (b, p)), v).$$

where φ was $\lambda\, \mathtt{isCenter}(a)$. This now satisfies the right definitional equality since if $p = \mathtt{refl}(a)$ and $b = a$, then $\mathtt{app}(\varphi, (a, \mathtt{refl}(a))) = \mathtt{refl}(a, \mathtt{refl}(a))$, and so using `subst` along a reflexivity path yields v.

3.5 Kan Completion

In this section we will show how to complete any cubical set Γ to a Kan cubical set $\hat{\Gamma}$. This works by freely attaching fillers to Γ; their restrictions are guided by the uniformity conditions. This construction however does not work for dependent types in a satisfactory way since it does not necessarily commute with substitutions.[2]

Theorem 3.24. *For any cubical set Γ there is a Kan cubical set $\hat{\Gamma}$ and a monomorphism* $\mathsf{inc}\colon \Gamma \to \hat{\Gamma}$ *such that for any Kan cubical set Δ and $\sigma\colon \Gamma \to \Delta$ there is a morphism $\hat{\sigma}\colon \hat{\Gamma} \to \Delta$ making the following diagram commute:*



Proof. Given a cubical set $\Gamma \vdash$ we define the sets $\hat{\Gamma}(I)$ for I in $\square$ and restriction maps $\hat{\Gamma}(I) \ni \rho \mapsto \rho f \in \hat{\Gamma}(J)$ for $f\colon I \to J$ by induction-recursion as follows. The sets are given by the rules:

1. If $\rho \in \Gamma(I)$, then $\mathsf{inc}\,\rho \in \hat{\Gamma}(I)$.
2. If S is an open box shape with principal side (x, a) and $\vec{u}$ is an S-open box in $\hat{\Gamma}(I)$, then $\mathsf{fill}_S\,\vec{u} \in \hat{\rho}$.
3. If S is an open box shape with principal side (x, a) and $\vec{u}$ is an S-open box in $\hat{\Gamma}(I)$ *and* $x = x_{I-x}$, then $\mathsf{comp}_S\,\vec{u} \in \hat{\Gamma}(I - x)$.

Here inc, fill_S, and comp_S are *constructors*, with the intended rôle for the latter two being the filling and composition operation, respectively. Note that in (2) and (3) being an open box refers to the restrictions defined at the same time. Note that the assumption on the variable x in (3) is due to the fact that x is bound. (One could also identify these expressions up to renaming of the bound variables, similar as for the path types.) The restrictions are guided by the uniformity conditions. For $f\colon I \to J$ we define

$$(\mathsf{inc}\,a)f = \mathsf{inc}(af)$$

$$(\mathsf{fill}_S\,\vec{u})f = \begin{cases} u_{yc}(f - y) & \text{if for some } (y, c) \in \langle S\rangle, fy = c, \\ \mathsf{comp}_{Sf'}(\vec{u}f') & \text{if } fx = a, \\ \mathsf{fill}_{Sf}(\vec{u}f) & \text{otherwise.} \end{cases}$$

[2] Indeed, type theory can't be consistently extended with such a rule, cf. `http://ncatlab.org/homotopytypetheory/show/Homotopy+Type+System#fibrant_replacement` (October 13, 2014).

where $f' = (f - x, x = x_J)$ and the restriction of a composition $\mathsf{comp}_S\,\vec{u}$ along $f\colon I - x \to J$ is defined by

$$(\mathsf{comp}_S\,\vec{u})f = \begin{cases} u_{yc}(x = a)(f - y) & \text{if for some } (y, c) \in \langle S \rangle, fy = c, \\ \mathsf{comp}_{S\tilde{f}}(\vec{u}\tilde{f}) & \text{otherwise.} \end{cases}$$

where now $\tilde{f} = (f, x = x_J)\colon I \to J, x_J$.

Now one proves by induction on $a \in \hat{\Gamma}(I)$ that $(\rho f)g = \rho(fg)$ and $\rho\mathbf{1} = \rho$ to make $\hat{\Gamma}$ into a cubical set. The Kan fillers are given by the constructor fill_S making $\hat{\Gamma}$ into a Kan cubical set. We directly get the monomorphism $\Gamma \to \hat{\Gamma}$ from the constructor inc.

Given a Kan cubical set Δ and $\sigma\colon \Gamma \to \Delta$ we define maps $\hat{\Gamma}(I) \to \Delta(I), \rho \mapsto \hat{\sigma}\rho$ while simultaneously proving $(\hat{\sigma}\rho)f = \hat{\sigma}(\rho f)$ for $f\colon I \to J$:

$$\begin{aligned} \hat{\sigma}(\mathsf{inc}\,\rho) &= \sigma\rho \\ \hat{\sigma}(\mathsf{fill}_S\,\vec{u}) &= [\Delta]_S(\hat{\sigma}\vec{u}) \\ \hat{\sigma}(\mathsf{comp}_S\,\vec{u}) &= |\Delta|_S(\hat{\sigma}\vec{u}) \end{aligned}$$

where (x, a) is the principle side of S (with $x = x_{I-x}$ in the last case), and $\hat{\sigma}\vec{u}$ is the family of elements $\hat{\sigma}(u_{yb})$ for $(y, b) \in \langle S \rangle$; by the induction hypothesis, this is an open box in $\Delta(I)$. That $\hat{\varphi}$ commutes with restrictions is by construction, as is the commuting diagram. $\square$

Chapter 4

The Universe of Kan Cubical Sets

In this chapter we will define the universe of small Kan types and show that it is itself a uniform Kan cubical set. The main work goes into the latter and we will decompose this into first defining the composition- and then the filling operations (similarly to what we did for Π-types).

Recall from Section 1.2.4 how to lift a Grothendieck universe $\mathbf{Set}_0$ to a universe in the presheaf model. This is adapted to give a universe of small Kan types by basically replacing "type" with "Kan type" in the definition: First, we adapt the definition of small type. The judgment $\Gamma \vdash A\ \mathsf{KType}_0$ is defined to be that $\Gamma \vdash A$ is a Kan type and $A\rho$ is a small set (i.e., an element of $\mathbf{Set}_0$) for each $\rho \in \Gamma(I)$; we also call such a $\Gamma \vdash A$ a *small Kan type*. Second, we define the universe accordingly:

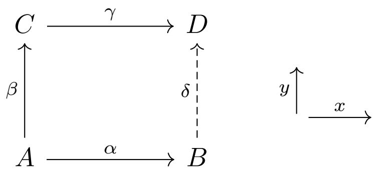
Definition 4.1. The cubical set $\mathbf{U}$ of small Kan types is defined as follows. The set $\mathbf{U}(I)$ consists of all small Kan types $\mathbf{y}I \vdash A\ \mathsf{KType}_0$; restrictions along $f\colon I \to J$ are defined by substituting with $\mathbf{y}f\colon \mathbf{y}J \to \mathbf{y}I$.

The definitions of $\ulcorner\cdot\urcorner$ and El are as in Section 1.2.4, where the fillings are defined by $[(\mathrm{El}\, T)\rho]_S\vec{u} = [(T\rho)_{\mathbf{1}}]_S\vec{u}$ and $[(\ulcorner A\urcorner\rho)_f]_S\vec{u} = [A(\rho f)]_S\vec{u}$. This defines a universe structure on the Kan cubical set model if we prove that $\Gamma \vdash \mathbf{U}$ is itself a Kan type, i.e., that $\mathbf{U}$ is a Kan cubical set.

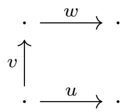
Note that the points of $\mathbf{U}$ are simply the (small) Kan cubical sets: for $\mathbf{y}\emptyset \vdash A$ we get a Kan cubical set with $A(I)$ being A_f where f is the unique $\emptyset \to I$. A line in $\mathbf{U}$ between points A and B can be seen as a "heterogeneous" notion of lines, cubes, …, $a \to b$ where a is an I-cube of A and b and I-cube of B.

Theorem 4.2. *The cubical set* $\mathbf{U}$ *of small Kan types is a Kan cubical set.*

We first show that $\mathbf{U}$ has compositions. The intuitive idea behind the composition is that of composing relations (hence the name). If we are given a open box in the universe, say of the form



we want to define the line δ in the universe. δ will be given by a family δ_f for $f\colon \{y\} \to I$ where the main case is $\delta_{\mathbf{1}}$ which will be defined to consist of triples (u, v, w) where $u \in \alpha_{\mathbf{1}}$, $v \in \beta_{\mathbf{1}}$, and $w \in \gamma_{\mathbf{1}}$ such that they are compatible in the sense that $u(x = 0) = v(y = 0)$ and $w(x = 0) = v(y = 1)$, i.e., a open box shape:



We then have to verify that δ has filling operations.

Lemma 4.3. $\mathbf{U}$ *has composition operations.*

Proof. Let $S = ((x, d); J; I)$ be an open box shape in I and $\vec{A}$ an S-open box in $\mathbf{U}(I)$, that is, compatible $A_{ya} \in \mathbf{U}(I - y)$ for $(y, a) \in \langle S\rangle$. We first define the composition $A_{xd} = |\mathbf{U}|_S \vec{A}$ as a type $\mathbf{y}(I - x) \vdash A_{xd}$ and then explain its Kan structure. Let $\bar{d} = 1 - d$.

Before we define A_{xd} let us introduce some notation. An S-open box $\vec{u}$ in $\vec{A}$ is given by a family $u_{yb} \in (A_{yb})_{\mathbf{1}}$ for $(y, b) \in \langle S\rangle$ such that they are compatible, i.e., $u_{yb}(z = c) = u_{zc}(y = b) \in (A_{yb})_{(z=c)} = (A_{zc})_{(y=b)}$ for $y \neq z$, $(y, b), (z, c) \in \langle S\rangle$. We denote the set of all such S-open boxes by $\langle S\rangle\vec{A}$. Note that if $f\colon I \to K$ is defined on x, J, then $\vec{u} \in \langle S\rangle\vec{A}$ implies $\vec{u}f \in \langle Sf\rangle\vec{A}f$.

For $f\colon I - x \to K$ we define the (small) set $(A_{xd})_f$ by distinguishing cases. In case $f(y) = b$ for some $(y, b) \in \langle S\rangle$ (note that $y \neq x$), then $f = (y = b)(f - y)$ and define $(A_{xd})_f = (A_{yb})_{(f-y,x=d)}$. Note that this is well defined as $\vec{A}$ is compatible. Otherwise, i.e., in case f is defined on J, we define $(A_{xd})_f = \langle Sf'\rangle Af'$ where $f' = (f, x = x_K)\colon I \to K, x_K$. This guarantees that A_{xd} has the right faces. One can also define it to be elements $\langle z\rangle\vec{u}$ with $\vec{u} \in \langle S(f, x = z)\rangle\vec{A}(f, x = z)$ and identify modulo α-conversion; we will use this notation.

To summarize, $(A_{xd})_f$ consists of elements of the form

1. $u \in (A_{yb})_{(f-y,x=d)}$ if $f(y) = b$ for some $(y, b) \in \langle S\rangle$;
2. $\langle z\rangle\vec{u}$, where $\vec{u} \in \langle S(f, x = z)\rangle\vec{A}(f, x = z)$ and z is fresh, otherwise.

Now if $g\colon K \to L$ we define the restrictions of an element in $(A_{xd})_f$ as follows. For elements u of the form (1), we use the restriction ug of $(A_{yb})_{(f-y,x=d)}$.

The restriction of an element $\langle z\rangle\vec{u}$ of the form (2) is defined by

$$(\langle z\rangle\vec{u})g = \begin{cases} u_{(fy)b}(g - fy, z = d) & \text{if } g(f(y)) = b \text{ for some } (y,b) \in \langle S\rangle, \\ \langle z'\rangle\vec{u}(g, z = z') & \text{otherwise,} \end{cases}$$

where z' is fresh w.r.t. the codomain of g. Note that in each case the resulting element is in $(A_{xd})_{fg}$. In particular, for $(y,b) \in \langle S\rangle$ and $f = \mathbf{1}$ we have

$$(\langle z\rangle\vec{u})(y = b) = u_{yb}(z = d).$$

This defines $A_{xd} = |\mathbf{U}|_S\vec{A}$ as a cubical set satisfying (as cubical sets)

$$(|\mathbf{U}|_S\vec{A})(y = b) = A_{yb} \qquad \text{for } (y,b) \in \langle S\rangle, \tag{4.1}$$

$$(|\mathbf{U}|_S\vec{A})f = |\mathbf{U}|_{Sf'}\vec{A}f' \qquad \text{if } f \text{ is defined on } J, \tag{4.2}$$

with $f' = (f, x = z)$ and in particular as sets we have

$$(|\mathbf{U}|_S\vec{A})_f = \begin{cases} (A_{yb})_{(f-y,x=d)} & \text{if } f(y) = b \text{ for some } (y,b) \in \langle S\rangle, \\ (|\mathbf{U}|_{Sf'}\vec{A}f')_{\mathbf{1}} & \text{if } f \text{ is defined on } J. \end{cases} \tag{4.3}$$

We now have to define the filling operations for $(A_{xd})_f = (|\mathbf{U}|_S\vec{A})_f$. W.l.o.g. we assume that $f = \mathbf{1}\colon I - x \to I - x$ as we take (4.3) as defining equations for the filling operations as well otherwise. Let $\vec{w}$ be an open box of shape $S' = ((x', d'); J'; I - x)$ in $(A_{xd})_{\mathbf{1}}$, i.e., $w_{yb} \in (A_{xd})_{(y=b)}$ for $(y,b) \in \langle S'\rangle$ such that for $(y,b), (z,c) \in \langle S'\rangle$, $y \neq z$

$$w_{yb}(z = c) = w_{zc}(y = b). \tag{4.4}$$

Note that for $(y,b) \in \langle S'\rangle - \langle S\rangle$ (i.e., $(y,b) \in \langle S'\rangle$ and $y \notin J$) we have that $w_{yb} = \langle x\rangle\vec{u}^{yb}$ with $\vec{u}^{yb} \in \langle S(y = b)\rangle\vec{A}(y = b)$. (We assume that all bound variables are x which is fresh for $I - x$.) Since $y \notin x, J$ we have $\langle S(y = b)\rangle = \langle S\rangle$, so $u^{yb}_{zc} \in (A_{zc})_{(y=b)}$ for $(z,c) \in \langle S\rangle$ such that

$$u^{yb}_{zc}(z' = c') = u^{yb}_{z'c'}(z = c) \tag{4.5}$$

whenever these elements are defined. Moreover, by the definition of the restriction

$$w_{yb}(z = c) = u^{yb}_{zc}(x = d) \tag{4.6}$$

and so, since $\vec{w}$ is adjacent compatible (4.4), we get for $(y,b) \in \langle S'\rangle - \langle S\rangle$ and $(z,c) \in \langle S\rangle$,

$$w_{zc}(y = b) = u^{yb}_{zc}(x = d). \tag{4.7}$$

Moreover, if $(y,b), (y',b') \in \langle S'\rangle - \langle S\rangle$ with $y \neq y'$, we have since $w_{yb}(y' = b') = w_{y'b'}(y = b)$ that the corresponding entries at $(z,c) \in \langle S\rangle$ of the vectors are equal, i.e.,

$$u^{yb}_{zc}(y' = b') = u^{y'b'}_{zc}(y = b). \tag{4.8}$$

For $(z, c) \in \langle S \rangle$ we denote the family (not necessarily an open box) of the u^{yb}_{zc}, $(y, b) \in \langle S' \rangle - \langle S \rangle$ by $\mathrm{aux}(z, c)$. By (4.8) this family is compatible.

We want to define the element $[(A_{xd})_{\mathbf{1}}]_{S'}\vec{w} = [(|\mathbf{U}|_S\vec{A})_{\mathbf{1}}]_{S'}\vec{w} \in (A_{xd})_{\mathbf{1}}$ in such a way that this definition satisfies the uniformity condition. I.e., for $f\colon I - x \to L$ defined on $x', J' \subseteq I - x$, we require

$$([(|\mathbf{U}|_S\vec{A})_{\mathbf{1}}]_{S'}\vec{w})f = [(|\mathbf{U}|_S\vec{A})_f]_{S'f}(\vec{w}f)$$

that is, according to (4.3),

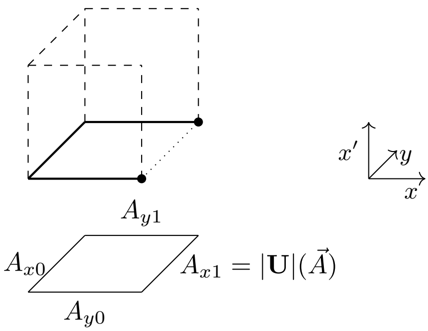
$$\big([(|\mathbf{U}|_S\vec{A})_{\mathbf{1}}]_{S'}\vec{w}\big)f = \begin{cases} [(|\mathbf{U}|_{Sf'}\vec{A}f')_{\mathbf{1}}]_{S'f}(\vec{w}f) & \text{if } J \subseteq \mathrm{def}(f), \\ [(A_{yb})_{(f-y,x=d)}]_{S'f}(\vec{w}f) & \text{if } fy = b \text{ for a } (y, b) \in \langle S \rangle. \end{cases} \tag{4.9}$$

Let us write $\langle x \rangle\vec{u}$ for the element $[(|\mathbf{U}|_S\vec{A})_{\mathbf{1}}]_{S'}\vec{w}$ we are going to define. To satisfy the second equation of (4.9) we need for $y \in J$ and $(y, b) \notin \langle S \rangle'$ that

$$u_{yb}(x = d) = (\langle x \rangle\vec{u})(y = b) = [(A_{yb})_{(x=d)}]_{S'(y=b)}(\vec{w}(y = b)). \tag{4.10}$$

To give $\langle x \rangle\vec{u}$ there are three cases to consider. We leave it for the reader to verify $(\langle x \rangle\vec{u})(y = b) = w_{yb}$ for $(y, b) \in \langle S \rangle$ along with the definition of $\vec{u}$.

1. *W.l.o.g.* $J \subseteq x', J'$. Let us first illustrate this in a (low-dimensional) special case where $I = \{x, x', y\}$, $J = \{y\}$, $d = d' = 1$, and also $J' = \emptyset$. We are given the dotted line in:



The types of the corresponding cubes are indicated in the lower square (which is not filled). The dotted line is, as an element in the composition, given by an open box indicated as the solid lines. To give the filling of the dotted line in the upwards direction is to give an open box indicated with the dashed lines; the first step is to fill each of the black dots upwards, and to proceed with the other cases with the extended box which now contains the non-principal sides for $y \in J$.

More formally and in the general case, for each $y \in J$ with $y \notin x', J'$ we have that $(y, b) \notin \langle S \rangle'$ (for $b \in \mathbf{2}$) and we construct

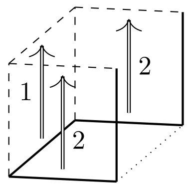
$$w_{yb} \in (A_{yb})_{(x=d)}$$

by filling $\vec{w}(y=b)$ of shape $S'(y=b)$ in $(A_{yb})_{(x=d)}$. Note that for $(y,b),(z,c)$ with $y,z\in J$ and $y,z\notin x',J'$ the so constructed elements are adjacent compatible since:

$$\begin{aligned} w_{yb}(z=c) &= ([(A_{yb})_{(x=d)}]_{S'(y=b)}(\vec{w}(y=b)))(z=c)\\ &= [(A_{yb})_{(x=d)(z=c)}]_{S'(y=b)(z=c)}(\vec{w}(y=b)(z=c))\\ &= [(A_{zc})_{(x=d)(y=b)}]_{S'(z=c)(y=b)}(\vec{w}(z=c)(y=b))\\ &= w_{zc}(y=b) \end{aligned}$$

Moreover, by construction they are adjacent compatible with the given open box $\vec{w}$. Thus, we can extend the $\vec{w}$ to a $((x,d);J',(J-(x',J'));I-x)$ open box.

2. *Case* $x'\notin J$. Let us first illustrate again in the special case as above but now with $J'=y$. So we are given the dotted line and the the solid lines on the right in:



The dotted line again corresponds to the three lower solid lines, and we want to construct three squares indicated by the dashed lines. To do so, we first fill on the left as indicated by the double arrow labeled "1"; second, we fill those sides labeled with "2" by the other sides taking into account those faces already constructed in the first step.

2.1. We construct $u_{x\bar{d}}\in(A_{x\bar{d}})_{\mathbf{1}}$ by filling $\mathrm{aux}(x,\bar{d})$ in $(A_{x\bar{d}})_{\mathbf{1}}$. Note that here $\mathrm{aux}(x,\bar{d})$ is an open box of shape $((x',d');J'-J;I-x)$.

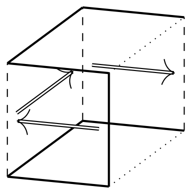
2.2. Next, for $(z,c)\in\langle S\rangle$ with $z\neq x$ we construct $u_{zc}\in(A_{x\bar{d}})_{\mathbf{1}}$ by filling in $(A_{x\bar{d}})_{\mathbf{1}}$ the open box

$$\mathrm{aux}(z,c),u_{x\bar{d}}(z=c),w_{zc}\text{ of shape }((x',d');(J'-J),x;I-z).$$

where the latter two elements are the non-principal sides at $(x,\bar{d})$ and (x,d), respectively.

2.3. This concludes the construction of $\vec{u}$ in this case.

3. *Case* $x'\in J$. Let us again first sketch the construction in the above special case where $x'=y$, $J=J'=y$. We are given the right hand open box given by the two dotted lines (which become the lower and upper solid lines) and the solid line on the right in:



The filler of the open box is now constructed doing the fillers indicated with the double arrows in order: starting with the front where there are three solid lines forming an open box, and continuing the way to the back, always taking into account the face of the previously constructed filler as principal side.

3.1. We begin by extending the input box along x in direction $\bar{d}$. More precisely, we construct $u_{zc} \in (A_{zc})_{\mathbf{1}}$ where $(z,c) \in \langle (x',d'); J - x' \rangle$ (note $J - x' = J \cap J'$) by filling

$$w_{zc}, \mathrm{aux}(z,c) \text{ of shape } ((x,\bar{d}); J' - J; I - z).$$

Here $w_{zc} \in (A_{xd})_{zc} = (A_{zc})_{(x=d)}$ is the principal side of the open box; moreover, note that $\langle (x',d'); J - x' \rangle$ contains $(x',\bar{d}')$ where $\bar{d}' = 1 - d'$, but not (x',d') and not $(x,\bar{d})$.

3.2. Next, we construct $u_{x\bar{d}} \in (A_{x\bar{d}})_{\mathbf{1}}$ by filling the open box

$$\begin{aligned} &u_{zc}(x = \bar{d}) \text{ for } (z,c) \in \langle (x',d'); J - x' \rangle, \\ &\mathrm{aux}(x,\bar{d}) \end{aligned}$$

of shape

$$((x',d'); (J - x') \cup (J' - J); I - x) = ((x',d'); J'; I - x) = S',$$

where $u_{x'\bar{d}'}(x = \bar{d}) \in (A_{x'\bar{d}'})_{(x=\bar{d})} = (A_{x\bar{d}})_{(x'=\bar{d}')}$ is the principal side of the open box.

3.3. Finally, we construct the missing side $u_{x'd'} \in (A_{x'd'})_{\mathbf{1}}$ by filling

$$\begin{aligned} &u_{x\bar{d}}(x' = d'), \\ &u_{zc}(x' = d') \text{ for } z \in J - x',\ c \in \mathbf{2}, \text{ and} \\ &\mathrm{aux}(x',d') \end{aligned}$$

of shape $((x,d); (J - x) \cup (J' - J); I - x') = ((x,d); J'; I - x')$.

3.4. This concludes the construction of $\vec{u}$ in this case.

We have to verify that this definition satisfies the uniformity conditions, i.e., that equations (4.9) are valid. Let $f \colon I - x \to L$ be defined on x', J'.

Assume $fy = b$ for some $y \in J$. To simplify notations, say $f = (z = b)$. Then $y \notin x', J'$ since f was defined on x', J'. Thus we obtain

$$(\langle x \rangle \vec{u})(y = b) = w_{yb} = [(A_{yb})_{(x=d)}]_{S'(x=d)}(\vec{w}(x = d))$$

as constructed in step 1, which we had to show.

Let us now assume that f is also defined on J. We have to show

$$(\langle x \rangle \vec{u}) f = [(|\mathbf{U}|_{Sf'} \vec{A} f')_{\mathbf{1}}]_{S'f}(\vec{w} f)$$

where $f' = (f, x = x^*)$ with x^* fresh. Let us denote the right hand side element by $\langle x^* \rangle \vec{u}^*$ and all abbreviations used in the definition of $\vec{u}^*$ will be decorated with a $*$ as well (e.g., aux^*). Thus we have to show $\vec{u} f' = \vec{u}^*$. Since f is injective $x' \in J$ iff $fx' \in fJ$, and thus $\vec{u}$ and $\vec{u}^*$ are defined via the same case. Moreover, for $(y, b) \in \langle S' \rangle - \langle S \rangle$ (which is iff $(fy, b) \in \langle S'f \rangle - \langle Sf' \rangle$) we have

$$w^*_{(fy)b} = w_{yb}(f - y) = (\langle x \rangle \vec{u}^{yb})(f - y) = \langle x^* \rangle\, \vec{u}^{yb}(f - y, x = x^*)$$

and thus $\mathrm{aux}(z, c)(f' - z) = \mathrm{aux}^*(f'z, c)$ for $(z, c) \in \langle S \rangle$. Now for example, in case 2 the first construction of $u^*_{(fx)\bar{d}}$ is by filling $\mathrm{aux}^*(fx, \bar{d})$, so

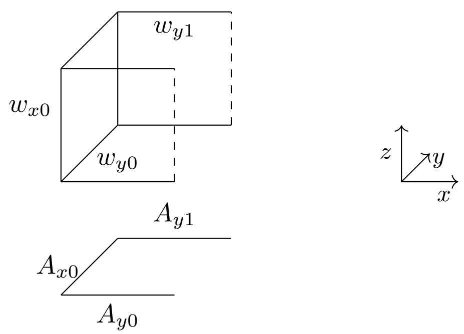
$$\begin{aligned} u^*_{x^*\bar{d}} &= [(A^*_{x^*\bar{d}})_{\mathbf{1}}](\mathrm{aux}^*(x^*, \bar{d})) \\ &= [(A^*_{x^*\bar{d}})_{\mathbf{1}}](\mathrm{aux}(x, \bar{d})(f' - x)) \\ &= [(A_{x\bar{d}})_f](\mathrm{aux}(x, \bar{d})f) \\ &= u_{x\bar{d}}f = u_{x\bar{d}}(f' - x) \end{aligned}$$

using the uniformity condition of $(A_{x\bar{d}})_{\mathbf{1}}$. Similarly, in the construction of the $u^*_{(fy)b}$ for $(y, b) \in \langle S \rangle$, $y \neq x$. The other case is analogous, concluding the proof. □

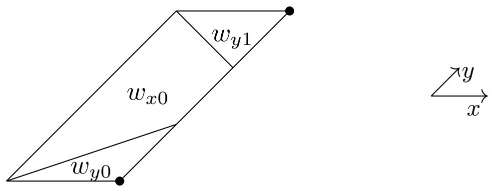
Theorem 4.4. $\mathbf{U}$ *is a Kan cubical set.*

Proof. We extend the composition operations of Lemma 4.3 to filling operations making $\mathbf{U}$ into a Kan cubical set. Let $S = ((x, d); J; I)$ be an open box shape in I and $\vec{A}$ an S-open box in $\mathbf{U}(I)$, i.e., adjacent compatible $A_{ya} \in \mathbf{U}(I - y)$ for $(y, a) \in \langle S \rangle$. We first define the filling $A = [\mathbf{U}]_S \vec{A}$ as a type $\mathbf{y}I \vdash A_{xd}$ and then explain its Kan structure. Note that this will be such that $A(x = d) = A_{xd}$ with $A_{xd} := |\mathbf{U}|_S \vec{A}$ as constructed in the preceding lemma.

For $f : I \to K$ we give a (small) set A_f. In case $fy = b$ for some $(y, b) \in \langle S \rangle \cup \{(x, d)\}$ we have $f = (y = b)(f - y)$ and set $A_f = (A_{yb})_{(f-y)}$. Otherwise, i.e., f is defined on x, J, we can w.l.o.g. assume $f = \mathbf{1} : I \to I$ as we otherwise set $A_f = ([\mathbf{U}]_{Sf} \vec{A}f)_{\mathbf{1}}$. The set $A_{\mathbf{1}}$ is now defined as follows: an element is of the form $\langle z \rangle \vec{w}$ where $\vec{w}$ is an open box in $\vec{A}s_z$ of shape Ss_z and z is fresh, so $\vec{w}$ is given by elements $w_{yb} \in (A_{yb})_{s_z}$ for $(y, b) \in \langle S \rangle$; moreover, we require for $(y, b) \in \langle S \rangle$ with $y \neq x$ that $w_{yb}(x = 1) \# z$, i.e., w_{yb} is degenerate along z. Here z is a bound variable. Let us illustrate this definition in the special case where $J = y$ and $d = 1$: such an element $\vec{w}$ is given by



where the dashed lines are required to be degenerate. Note that projecting $\vec{w}$ to $(z = 1)$ gives an element of $A_{x1} = |\mathbf{U}|(\vec{A})$ (disregarding binders for the moment). A good way to think of $\vec{w}$ is to imagine w_{y0} and w_{y1} as triangles by shrinking each of the dashed lines to a point, and to think of the dimension z as "hidden":



Here, the right hand lines are the projection $(z = 1)$, which is how we will define the restriction of the element $\vec{w}$ to $(x = 1)$, and the dots correspond to the dashed lines above.

The restriction $ug \in A_{fg}$ along $g\colon K \to L$ of an element $u \in A_f$ where $f\colon I \to K$ is defined as follows: In case $fy = b$ for some $(y, b) \in \langle S \rangle \cup \{(x, d)\}$, ug is given by the restriction of $(A_{yb})_{(f-y)}$. Otherwise, f is defined on x, J and $u = \langle z \rangle \vec{w}$ with $\vec{w}$ and open box in $\vec{A}f$ of shape $(Sf)s_z$; we set

$$(\langle z \rangle \vec{w})g = \begin{cases} w_{(fy)b}(z = 0)(g - fy) & \text{if } g(fy) = b \text{ for some } (y, b) \in \langle S \rangle, \\ (\langle x \rangle \vec{w}(z = 1))(g - fx) & \text{if } g(fx) = d \text{ and } fJ \subseteq \mathrm{def}(g), \\ \langle z' \rangle\, \vec{w}(g, z = z') & \text{otherwise,} \end{cases}$$

where in the last case z' is fresh w.r.t. the codomain of g. In particular, if $f = \mathbf{1}$, this definitions reads as

$$(\langle z \rangle \vec{w})g = \begin{cases} w_{yb}(z = 0)(g - y) & \text{if } gy = b \text{ for some } (y, b) \in \langle S \rangle, \\ (\langle x \rangle \vec{w}(z = 1))(g - x) & \text{if } gx = d \text{ and } J \subseteq \mathrm{def}(g), \\ \langle z' \rangle\, \vec{w}(g, z = z') & \text{otherwise.} \end{cases}$$

This definition deserves some explanation: in case $gy = b$ for $(y, b) \in \langle S \rangle$, $w_{yb} \in (A_{yb})_{s_z}$ and thus $w_{yb}(z = 0)(g - y) \in (A_{yb})_{(g-y)} = A_g$; in the second case where $gx = d$, $\vec{w}(z = 1)$ is an S-open box in $\vec{A}$, and thus, $\langle x \rangle \vec{w}(z = 1)$ is an an element of $(A_{xd})_{\mathbf{1}} = (|\mathbf{U}|_S \vec{A})_{\mathbf{1}}$ (cf. the definition in Lemma 4.3). It can be checked that this defines a (small) type $\mathbf{y}I \vdash [\mathbf{U}]_S \vec{A} = A$ satisfying (as types, not yet as Kan types)

$$\begin{aligned} ([\mathbf{U}]_S \vec{A})(y = b) &= A_{yb} && \text{for } (y, b) \in \langle S \rangle \\ ([\mathbf{U}]_S \vec{A})(x = d) &= |\mathbf{U}|_S \vec{A} \\ ([\mathbf{U}]_S \vec{A})f &= [\mathbf{U}]_{Sf} \vec{A}f && \text{if } f \text{ is defined on } x, J. \end{aligned}$$

The next step is to define the Kan structure on A_f where $f\colon I \to K$. W.l.o.g. we assume that $f = \mathbf{1}\colon I \to I$ as otherwise we use the above equations for the filling. Let $\vec{v}$ be an open box of shape $S' = ((x', d'); J'; I)$ in $A = [\mathbf{U}]_S \vec{A}$, i.e., $\vec{v}$ is given by adjacent-compatible $v_{yb} \in A_{(y=b)}$ for $(y, b) \in \langle S' \rangle$.

Note that if $(y, b) \in \langle S' \rangle \cap \langle S \rangle$, we have $v_{yb} \in A_{(y=b)} = (A_{yb})_{\mathbf{1}}$.

For $(y, b) \in \langle S' \rangle - \langle S \rangle$, we have $v_{yb} \in A_{(y=b)} = ([\mathbf{U}]_{S(y=b)} \vec{A}(y = b))_{\mathbf{1}}$, and so $v_{yb} = \langle z \rangle \vec{w}^{yb}$ where $\vec{w}^{yb}$ is an $S(y = b)s_z$ open box in $\vec{A}(y = b)s_z$ with the conditions described above. In particular, $w^{yb}_{\xi c} \in (A_{\xi c})_{(y=b)s_z}$ for $(\xi, c) \in \langle S \rangle$ with

$$w^{yb}_{\xi c}(\xi' = c') = w^{yb}_{\xi' c'}(\xi = c) \tag{4.11}$$

if also $(\xi', c') \in \langle S \rangle$ with $\xi \neq \xi'$. Moreover, since $\vec{v}$ is adjacent compatible we get that $v_{yb}(y' = b') = v_{y'b'}(y = b)$ for both $(y, b), (y', b') \in \langle S' \rangle - \langle S \rangle$, and thus the corresponding entries at position $(\xi, c) \in \langle S \rangle$ of the vectors $\vec{w}^{yb}$ and $\vec{w}^{y'b'}$ coincide, i.e.,

$$w^{yb}_{\xi c}(y' = b') = w^{y'b'}_{\xi c}(y = b). \tag{4.12}$$

Similar to the preceding lemma, for $(\xi, c) \in \langle S \rangle$ the adjacent-compatible family (not necessarily open box) of the $w^{yb}_{\xi c} \in (A_{\xi c})_{s_z(y=b)}$ for $y \in J' - (J, x)$ and $b \in \mathbf{2}$ is denoted by $\operatorname{aux}(\xi, c)$. Note that $\operatorname{aux}(\xi, c)$ is only defined on $J' - (J, x)$ and *not* on $\langle S' \rangle - \langle S \rangle$.

We want to construct $[A_{\mathbf{1}}]_{S'} \vec{v} = [([\mathbf{U}]_S \vec{A})_{\mathbf{1}}]_{S'} \vec{v} \in A_{\mathbf{1}}$ in a uniform way so that it satisfies for $f \colon I \to K$ defined on x', J'

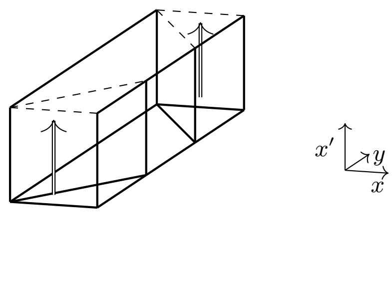
$$\big([([\mathbf{U}]_S \vec{A})_{\mathbf{1}}]_{S'} \vec{v}\big) f = [([\mathbf{U}]_S \vec{A})_f]_{S'f}(\vec{v} f)$$

that is,

$$\big([([\mathbf{U}]_S \vec{A})_{\mathbf{1}}]_{S'} \vec{v}\big) f = \begin{cases} [(A_{yb})_{(f-y)}]_{S'}(\vec{v} f) & \text{if } fy = b \text{ for some} \\ & (y, b) \in \{(x, d)\} \cup \langle S \rangle, \\ [([\mathbf{U}]_{Sf} \vec{A} f)_{\mathbf{1}}]_{S'f}(\vec{v} f) & \text{otherwise.} \end{cases} \tag{4.13}$$

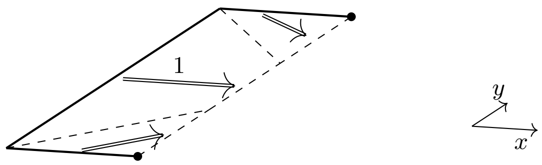
The element $[A_{\mathbf{1}}]_{S'} \vec{v}$ will be given by $\langle z \rangle \vec{w}$ with $\vec{w}$ an Ss_z open box in $\vec{A} s_z$ with the above provisos. The construction distinguishes several cases where in each case we assume that the previous cases didn't apply. We set $\bar{d} = 1 - d$ and $\bar{d}' = 1 - d'$.

1. *W.l.o.g.* $x, J \subseteq x', J'$. For each $y \in x, J$ with $y \notin x', J'$ we extend the input box $\vec{v}$ with $v_{yb} \in (A_{yb})_{s_z}$ $(b \in \mathbf{2})$ constructed as follows: v_{yb} is the filler in $(A_{yb})_{s_z}$ of the open box given by $\vec{v}(y = b)$ of shape $S'(y = b)$ in $(A_{yb})_{s_z}$. The final result will be an open box of shape $((x', d); J' \cup ((x, J) - (x', J')); I)$; to check that the so added sides indeed are adjacent compatible is similar to the verification done in the proof of the preceding lemma. Let us illustrate this in the special case with $J = y$, $J' = x$, and $x' \notin x, J$ (and $d = d' = 1$). We are given the sides enclosed by the solid lines and want to fill the whole shape:



In the picture we are hiding the extra dimension as discussed above. The input box is extended with the non-principal sides for y by filling the sides indicated with the double arrows.

2. *Case* $x = x'$ and $d = d'$. Then $(x, \bar{d}) \in \langle S' \rangle$. A simple special case like above but with $J = J' = y$ can be illustrated by:



Here the algorithm proceeds by first filling along the double arrow labeled "1", and then filling along the other arrows taking the side constructed in the first filling into account as a non-principal side (and where the opposing sides are the respective degenerates of the indicated points).

2.1. First, we construct $w_{x\bar{d}} \in (A_{x\bar{d}})_{s_z}$ by filling the open box

$$\begin{aligned} &v_{x\bar{d}} \in (A_{x\bar{d}})_{\mathbf{1}}, \\ &\mathrm{aux}(x, \bar{d}). \end{aligned}$$

Note that $\mathrm{aux}(x, \bar{d})$ is defined on the sides of $J' - (J, x)$, which is $J' - J$ in this case. Thus the open box has shape $((z, 1); J' - J; (I - x), z)$.

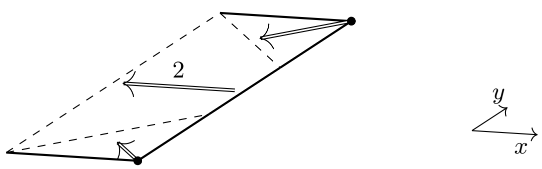
2.2. Next, we construct $w_{yb} \in (A_{yb})_{s_z}$ for $y \in J$ by filling the open box

$$v_{yb} \in A_{(y=b)} = (A_{yb})_{\mathbf{1}}, \tag{4.14}$$
$$v_{yb}(x = d)s_z \in (A_{yb})_{s_z(x=d)}, \tag{4.15}$$
$$v_{x\bar{d}}(y = b) \in (A_{x\bar{d}})_{s_z(y=b)} = (A_{yb})_{s_z(x=\bar{d})} \tag{4.16}$$
$$\mathrm{aux}(y, b)$$

which is of shape $((z, 1); x, (J' - J); (I - y), z)$. Here (4.14) is the principal side, (4.15) is at the non-principal side (x, d), and (4.16) is at the $(x, \bar{d})$ side.

2.3. This concludes the construction of $\vec{w}$.

3. *Case* $x = x'$ and $d = 1 - d' = \bar{d}'$. Then the element $v_{xd} \in A_{(x=d)} = (A_{xd})_{\mathbf{1}} = (|\mathbf{U}|_S \vec{A})_{\mathbf{1}}$ is of the form $v_{xd} = \langle x \rangle \vec{u}$ where $\vec{u}$ is an S open box in $\vec{A}$ by definition of the compositions in the previous lemma. In the same special case as above, the situation can be depicted as:



Now the order of the filling is reversed: first fill the "triangles" as indicated, and then along the arrow labeled "2", taking into account the already constructed sides.

3.1. First, for each $(y, b) \in J \times \mathbf{2}$ we construct w_{yb} by filling

$$
\begin{aligned}
&u_{yb}(x = d')s_z, && (4.17)\\
&v_{yb} \in A_{(y=b)} = (A_{yb})_{\mathbf{1}}, && (4.18)\\
&u_{yb} \in (A_{yb})_{\mathbf{1}}, && (4.19)\\
&\operatorname{aux}(y, b)
\end{aligned}
$$

in $(A_{yb})_{s_z}$ of shape $((x, \bar{d}); z, (J' - J); (I - y), z)$. Here, (4.17) is the principal face, and (4.18) and (4.19) are the non-principal faces at $(z, 0)$ and $(z, 1)$, respectively.

3.2. Next, we construct $w_{x\bar{d}} \in (A_{x\bar{d}})_{s_z}$ by filling the open box given by

$$
\begin{aligned}
&u_{x\bar{d}} \in (A_{x\bar{d}})_{\mathbf{1}}, \\
&w_{yb}(x = \bar{d}) \in (A_{yb})_{s_z(x=\bar{d})} && \text{for } (y, b) \in J \times \mathbf{2}, \\
&\operatorname{aux}(x, \bar{d}) && \text{at all sides of } J' - J
\end{aligned}
$$

of shape $((z, 0); J \cup (J' - J); (I - x), z) = ((z, 0); J'; (I - x), z)$.

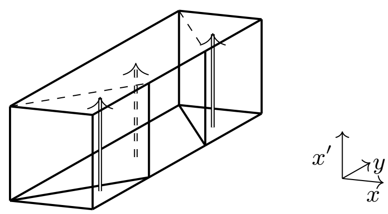
3.3. This concludes the construction of $\vec{w}$.

4. *Case* $x' \notin J$. As in the previous case, the element $v_{xd} \in A_{(x=d)} = (A_{xd})_{\mathbf{1}} = (|\mathbf{U}|_S \vec{A})_{\mathbf{1}}$ is of the form $v_{xd} = \langle x \rangle \vec{u}$ where $\vec{u}$ is an S-open box in $\vec{A}$. Moreover, $v_{x'\bar{d}'} \in A_{(x'=\bar{d}')} = ([\mathbf{U}]_S \vec{A})_{(x'=\bar{d}')}$ and since $x' \notin J$ and $x \neq x'$, so $(x', \bar{d}') \in \langle S' \rangle - \langle S \rangle$,

$$([\mathbf{U}]\vec{A})_{(x'=\bar{d}')} = ([\mathbf{U}]\vec{A}(x' = \bar{d}'))_{\mathbf{1}},$$

and so $v_{x'\bar{d}'} = \langle z \rangle \vec{w}^{x'\bar{d}'}$ with $\vec{w}^{x'\bar{d}'}$ an open box in $\vec{A}(x' = \bar{d}')$ (of shape $S(x' = \bar{d}')s_z$).

Let us again illustrate a special case with $J = y$ and $J' = x, y$. We are given those sides enclosed by solid lines and the top faces are missing:



Here the algorithm proceeds by first filling the middle cube along the dashed double arrow, and then the other two cubes double arrows taking into account the faces of the constructed middle cube (with opposing non-principle side given by degenerates).

4.1. First, we construct $w_{x\bar{d}} \in (A_{x\bar{d}})_{s_z}$ by filling the open box

$w^{x'\bar{d}'}_{x\bar{d}} \in (A_{x\bar{d}})_{(x'=\bar{d}')}$	as principal side,
$v_{x\bar{d}} \in A_{(x=\bar{d})} = (A_{x\bar{d}})_{\mathbf{1}}$	at side $(z, 0)$,
$u_{x\bar{d}} \in (A_{x\bar{d}})_{\mathbf{1}}$	at side $(z, 1)$,
$\mathrm{aux}(x, \bar{d})$	at all sides of $J' - (x, J)$,

which is of shape $((x', d'); z, (J' - (J, x)); (I - x), z)$.

4.2. Next, we construct the other $w_{yb} \in (A_{yb})_{s_z}$ for $(y, b) \in J \times \mathbf{2}$ by filling

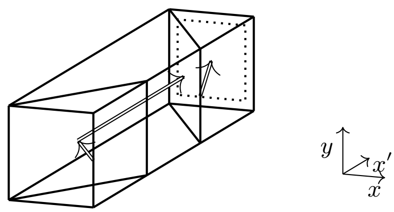
$w^{x'\bar{d}'}_{yb} \in (A_{yb})_{(x'=\bar{d}')}$	as principal side,
$v_{yb} \in A_{(y=b)} = (A_{yb})_{\mathbf{1}}$	at side $(z, 0)$,
$u_{yb} \in (A_{yb})_{\mathbf{1}}$	at side $(z, 1)$,
$w_{x\bar{d}}(y = b) \in (A_{x\bar{d}})_{s_z(y=b)} = (A_{yb})_{s_z(x=\bar{d})}$	at side $(x, \bar{d})$,
$u_{yb}(x = d)s_z \in (A_{yb})_{s_z(x=d)}$	at side (x, d),
$\mathrm{aux}(y, b)$	at all sides of $J' - (x, J)$

in $(A_{yb})_{s_z}$ which is of shape $((x', d'); z, x, (J' - (x, J)); (I - y), z)$.

4.3. This concludes the construction of $\vec{w}$.

5. *Case* $x' \in J$. As in the previous cases, the element $v_{xd} \in A_{(x=d)} = (A_{xd})_{\mathbf{1}} = (|\mathbf{U}|_S \vec{A})_{\mathbf{1}}$ is of the form $v_{xd} = \langle x \rangle \vec{u}$ where $\vec{u}$ is an S-open box in $\vec{A}$.

A special case (with $J = x'$ and $J' = x, y$) of this situation can depicted as a hollow box where the face indicated with dots is missing:



Here the algorithm proceeds in three steps: first filling the "prism" opposed to the dotted square (with principal face being the invisible degenerate square), then filling the middle cube, and finally filling the prism touching the dotted square.

5.1. First, we construct $w_{yb} \in (A_{yb})_{s_z}$ for $(y, b) \in \langle (x', d'); J - x' \rangle$ by filling the open box

$u_{yb}(x = d)s_z \in (A_{yb})_{s_z(x=d)}$	as principal side (x, d),
$v_{yb} \in A_{(y=b)} = (A_{yb})_{\mathbf{1}}$	at side $(z, 0)$,
$u_{yb} \in (A_{yb})_{\mathbf{1}}$	at side $(z, 1)$,
$\mathrm{aux}(y, b)$	at all sides of $J' - (x, J)$

of shape $((x,\bar{d}); z, (J' - (x,J)); (I - y), z)$.

5.2. Second, we construct $w_{x\bar{d}} \in (A_{x\bar{d}})_{s_z}$ by filling the following open box: the elements constructed so far induce an open box

$$\begin{aligned} & w_{yb}(x = \bar{d}) \in (A_{yb})_{s_z(x=\bar{d})} = (A_{x\bar{d}})_{s_z(y=b)} \\ & \qquad \text{for } (y,b) \in \langle (x',d'); J - x'; z, (I - (x,y)) \rangle \end{aligned}$$

which we extend to an open box by adding the non-principal faces

$$\begin{array}{ll} v_{x\bar{d}} \in A_{(x=\bar{d})} = (A_{x\bar{d}})_{\mathbf{1}} & \text{at side } (z,0), \\ u_{yb} \in (A_{yb})_{\mathbf{1}} & \text{at side } (z,1), \\ \mathrm{aux}(y,b) & \text{at all sides of } J' - (x,J), \end{array}$$

to obtain an open box in $(A_{x\bar{d}})_{s_z}$ of shape $((x',d'); (J - x'), z, (J' - J); (I - x), z)$.

5.3. Last, we construct the missing $w_{x'd'} \in (A_{x'd'})_{s_z}$ by filling the open box given by

$$\begin{array}{ll} u_{x'd'} \in (A_{x'd'})_{\mathbf{1}} & \text{as principal side at } (z,1), \\ w_{x\bar{d}}(x' = d') \in (A_{x\bar{d}})_{s_z(x'=d')} = (A_{x'd'})_{s_z(x=\bar{d})} & \text{at side } (x,\bar{d}), \\ u_{x'd'}(x = d)s_z & \text{at side } (x,d), \\ \mathrm{aux}(x',d') & \text{at all sides of } J' - (J,x), \\ w_{yb}(x' = d') \in (A_{yb})_{s_z(x'=d')} = (A_{x'd'})_{s_z(y=b)} & \text{for } (y,b) \text{ with } y \in J - x', \end{array}$$

which has shape $((z,0); x, (J' - (J,x)), (J - x'); (I - x'), z)$.

5.4. This concludes the construction of $\vec{w}$.

The lengthy verification of the uniformity conditions is similar as sketched in Lemma 4.3 and is omitted. □

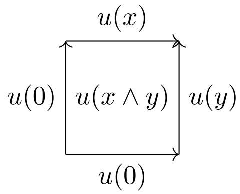
Conclusion

Let us conclude this thesis by summarizing what has been done and indicate future directions of research. We have given a model of dependent types based on a notion of cubical sets in a *constructive meta theory.* This model supports dependent products and sums, identity types, and universes. To give the interpretation of types we have to require a so-called uniform Kan structure which is a refinement of Kan's original extension condition on cubical sets; this condition is natural given the interpretation of the cubical set operations as "substitution operations".

One aspect not discussed in this thesis is the implementation [11] based on (a nominal variation of) the Kan cubical set model presented here. To sketch the basic idea, we start with type theory without identity types plus primitive notions such as `Id`, `refl`, `subst`, `substEq`, `isCenter`, and `funExt`. (The implementation also supports primitives which entail the univalence axiom.) There is an evaluation of terms (which may dependent on the just mentioned primitives) into values. Each value depends on finitely many names and there are the basic operations of cubical sets on values: we can rename a name into a fresh name or take a face. Values reflect the constructions in the model; e.g., there is path abstraction $\langle x \rangle u$ and filling operations. During evaluation names are introduced, e.g., $\mathtt{refl}\, a$ is evaluated to $\langle x \rangle u$ where u is the value of a and x is a fresh name for u (in the implementation degeneracy maps are implicit). Moreover, the primitives are evaluated to values like it was described in this thesis. Naturally, we also have to explain the Kan structure operations for each type. During type-checking the aforementioned primitives are treated as uninterpreted constants. But whenever the type-checker has to check for conversion of two terms this is done by *evaluating* their two values and comparing those values.

As we have seen, in the model the usual equation for the `J`-eliminator holds only up to propositional equality and *not* definitional equality. Restricting to regular Kan types (cf. Section 3.4) allows for definitional equality; this regularity is, however, not closed under function types. In a recent variation [14] of the cubical set model covered in this thesis, regularity is however preserved by all type formers. The main difference to the present model is to use a variation of cubical sets: these are also equipped with the so-called *connections* which correspond to operations $x \wedge y$ and $x \vee y$ on names, satisfying the rules of a (bounded) distributive lattice; for cubical sets this allows, e.g., given a

$u(x)$ depending on the name x to form the square (leaving degeneracy maps implicit):



This square is such that if u degenerate along x the square is (the degenerate of) $u(0)$. Moreover, this variation of cubical sets also allows taking *diagonals*, i.e., if a u depends on a name x there is an operation $u(x = y)$ even if u itself depends on y! Another difference to the present model is in the Kan structure: one only requires composition operations satisfying regularity (the filling operations can be derived from those with the help of connections) but on more general "open box shapes". (One can also add symmetries $1-x$ so that the structure on names becomes that of a *De Morgan algebra*.) This model has also been implemented and extended with an (experimental) implementation of higher inductive types [11][1].

Another direction of future research is to explore the relations with the more categorically formulated model constructions using the notion of weak factorization systems. Since we don't model the definitional equality of `J` here, it is clear that the model presented here has no underlying factorization system. Ongoing work [41] aims to give a variation of the current model using algebraic weak factorization system (also in a constructive meta theory). Especially some aspects of the variation of the model mentioned in the previous paragraph has a lot of resemblance with the work on path object categories by van den Berg and Garner [43]; [16] shows that cubical sets with connections form an instance of such a path object category.

Another direction of work is to formulate a *cubical type theory*, i.e., a type theory where one can directly argue about and manipulate the (hyper) cube structure. In such a theory, names (and related concepts like name/path abstraction and application) should be first-class entities, as well as the Kan structure should be exposed to the users. Such a system formulates typing rules underlying the values from the implementations mentioned above. The uninterpreted constants from before, like function extensionality, can then directly be *implemented* inside this cubical type theory. Another aspect of directly being able to manipulate higher-dimensional cubes is that it allows for simpler proofs when doing synthetic homotopy theory inside type theory [30]. An implementation of such a cubical type theory is ongoing work [12]. A similar such "enriched" type theory for internalized parametricity was recently given in [5] (based on a presheaf model similar to the one considered here). Similar type theories have been proposed by Altenkirch and Kaposi [1], Polonsky [39], and Brunerie and Licata [8].

[1] On the branch `connections` dated March 19, 2015

Bibliography

[1] Thorsten Altenkirch and Ambrus Kaposi, *Towards cubical type theory*, Preprint, 2014.

[2] Thorsten Altenkirch, Conor McBride, and Wouter Swierstra, *Observational equality, now!*, PLPV '07: Proceedings of the 2007 workshop on Programming languages meets program verification (New York, NY, USA), ACM, 2007, pp. 57–68.

[3] Steve Awodey and Michael A. Warren, *Homotopy theoretic models of identity types*, Math. Proc. Cambridge Philos. Soc. **146** (2009), no. 1, 45–55.

[4] Bruno Barras, Thierry Coquand, and Simon Huber, *A generalization of the Takeuti-Gandy interpretation*, To appear in Mathematical Structures in Computer Science, 2014.

[5] Jean-Philippe Bernardy, Thierry Coquand, and Guilhem Moulin, *A presheaf model of parametric type theory*, Preprint, 2015.

[6] Marc Bezem and Thierry Coquand, *A Kripke model for simplicial sets*, Theoretical Computer Science **574** (2015), 86–91.

[7] Marc Bezem, Thierry Coquand, and Simon Huber, *A model of type theory in cubical sets*, 19th International Conference on Types for Proofs and Programs (TYPES 2013) (Dagstuhl, Germany) (Ralph Matthes and Aleksy Schubert, eds.), Leibniz International Proceedings in Informatics (LIPIcs), vol. 26, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2014, pp. 107–128.

[8] Guillaume Brunerie and Dan Licata, *A cubical infinite-dimensional type theory*, Slides of a talk at Oxford Homotopy Type Theory Workshop, November 2014.

[9] John Cartmell, *Generalised algebraic theories and contextual categories*, Annals of Pure and Applied Logic **32** (1986), 209–243.

[10] Alonzo Church, *A formulation of the simple theory of types*, The Journal of Symbolic Logic **5** (1949), no. 2, 56–68.

[11] Cyril Cohen, Thierry Coquand, Simon Huber, and Anders Mörtberg, *Cubical*, `https://github.com/simhu/cubical`, 2013.

[12] ———, *Cubicaltt*, `https://github.com/mortberg/cubicaltt`, 2015.

[13] Thierry Coquand, *A property of contractible types*, Unpublished note available at `http://www.cse.chalmers.se/~coquand/contr.pdf`.

[14] ———, *Variation on cubical sets*, Unpublished note available at `http://www.cse.chalmers.se/~coquand/comp.pdf`, 2014.

[15] Thierry Coquand and Nils Anders Danielsson, *Isomorphism is equality*, Indagationes Mathematicae **24** (2013), no. 4, 1105–1120.

[16] Simon Docherty, *A model of type theory in cubical sets with connections*, Master's thesis, Universiteit van Amsterdam, 2014.

[17] Peter Dybjer, *Internal type theory*, Lecture Notes in Computer Science, Springer Verlag, Berlin, Heidelberg, New York, 1996, pp. 120–134.

[18] Richard Garner, *Factorisation axioms for type theory*, Notes for a talk given at the Workshop "Identity Types – Topological and Categorical Structure" (Uppsala, November 13-14, 2006), available at `http://comp.mq.edu.au/~rgarner/Papers/Uppsala.pdf`, 2006.

[19] Paul G. Goerss and John F. Jardine, *Simplicial homotopy theory*, Progress in Mathematics, no. 174, Birkhäuser Basel, 1999.

[20] Georges Gonthier, *Formal proof–the four-color theorem*, Notices of the AMS **55** (2008), no. 11, 1382–1393.

[21] Georges Gonthier, Andrea Asperti, Jeremy Avigad, Yves Bertot, Cyril Cohen, François Garillot, Stéphane Le Roux, Assia Mahboubi, Russell O'Connor, Sidi Ould Biha, Ioana Pasca, Laurence Rideau, Alexey Solovyev, Enrico Tassi, and Laurent Théry, *A Machine-Checked Proof of the Odd Order Theorem*, ITP 2013, 4th Conference on Interactive Theorem Proving (Rennes, France) (Sandrine Blazy, Christine Paulin, and David Pichardie, eds.), Lecture Notes in Computer Science, vol. 7998, Springer, July 2013, pp. 163–179.

[22] Marco Grandis and Luca Mauri, *Cubical sets and their site*, Theory and Applications of Categories **11** (2003), no. 8, 185–211.

[23] Alexandre Grothendieck, *Pursuing stacks*, Manuscript, 1983.

[24] Martin Hofmann, *Extensional concepts in intensional type theory*, Ph.D. thesis, University of Edinburgh, 1995.

[25] ———, *Syntax and semantics of dependent types*, Semantics and logics of computation (A.M. Pitts and P. Dybjer, eds.), Publ. Newton Inst., vol. 14, Cambridge University Press, Cambridge, 1997, Papers from the Summer

School held at the University of Cambridge, Cambridge, September 1995, pp. 79–130.

[26] Martin Hofmann and Thomas Streicher, *Lifting Grothendieck universes*, Unpublished Note.

[27] ———, *The groupoid interpretation of type theory*, Twenty-five years of constructive type theory (Venice, 1995), Oxford Logic Guides, vol. 36, Oxford University Press, New York, 1998, pp. 83–111.

[28] Daniel M. Kan, *Abstract homotopy. I*, Proceedings of the National Academy of Sciences of the United States of America **41** (1955), no. 12, 1092–1096.

[29] Chris Kapulkin, Peter LeFanu Lumsdaine, and Vladimir Voevodsky, *The simplicial model of univalent foundations*, Preprint, `http://arxiv.org/abs/1211.2851`, 2012.

[30] Dan Licata and Guillaume Brunerie, *A cubical approach to synthetic homotopy theory*, Preprint, 2015.

[31] Per Martin-Löf, *An intiutionistic theory of types: Predicative part*, Logic Colloquium '73 (H. E. Rose and J. Shepherdson, eds.), North–Holland, Amsterdam, 1975, pp. 73–118.

[32] ———, *Intuitionistic type theory*, Bibliopolis, 1984.

[33] J. Peter May, *Simplicial objects in algebraic topology*, Van Nostrand Mathematical Studies, vol. 11, Van Nostrand, 1967.

[34] Thomas Nikolaus, *Algebraic models for higher categories*, Indagationes Mathematicae **21** (2011), no. 1–2, 52–75.

[35] Christine Paulin-Mohring, *Inductive Definitions in the System Coq - Rules and Properties*, Proceedings of the conference Typed Lambda Calculi and Applications (Marc Bezem and Jan Friso Groote, eds.), Lecture Notes in Computer Science, no. 664, 1993.

[36] Andrew M. Pitts, *An equivalent presentation of the Bezem-Coquand-Huber category of cubical sets*, Preprint arXiv:1401.7807 [cs.LO], December 2013.

[37] ———, *Nominal sets: Names and symmetry in computer science*, Cambridge Tracts in Theoretical Computer Science, vol. 57, Cambridge University Press, 2013.

[38] ———, *Nominal presentation of the cubical sets model of type theory*, Preprint, August 2014.

[39] Andrew Polonsky, *Internalization of extensional equality*, Preprint arXiv:1401.1148v3 [cs.LO], January 2015.

[40] Ross Street, *Cosmoi of internal categories*, Transactions of the American Mathematical Society **258** (1980), no. 2, 271–318.

[41] Andrew Swan, *An algebraic weak factorisation system on 01-substitution sets: A constructive proof*, Preprint arXiv:1409.1829 [math.LO], September 2014.

[42] The Univalent Foundations Program, *Homotopy type theory: Univalent foundations of mathematics*, `http://homotopytypetheory.org/book`, Institute for Advanced Study, 2013.

[43] Benno van den Berg and Richard Garner, *Topological and simplicial models of identity types*, Transactions of the ACM on Computational Logic **13** (2012), no. 1, 3:1–3:44.