Cartesian Cubical Type Theory

Carlo Angiuli, Guillaume Brunerie, Thierry Coquand,
Kuen-Bang Hou (Favonia), Robert Harper, Daniel R. Licata

Draft as of 12/5/2017

Abstract

We present a cubical type theory based on the Cartesian cube category (faces, degeneracies, symmetries, diagonals, but no connections or reversal) with univalent universes, each containing Π, Σ, path, identity, natural number, boolean, pushout, and glue (equivalence extension) types. The type theory includes a syntactic description of a uniform Kan operation, along with judgemental equality rules defining the Kan operation on each type. The Kan operation uses both a different set of trivial cofibrations and a different set of cofibrations than the Cohen, Coquand, Huber, and Mörtberg (CCHM) model.

Next, we describe a constructive model in Cartesian cubical sets; the syntactic type theory is inspired by this model, though we have not yet given a formal interpretation. We describe a mechanized proof, using the internal language of cubical sets in the style introduced by Orton and Pitts, that glue, Π, Σ, path, identity, boolean, natural number, and pushout types are Kan in this model; we also sketch a proof that this internal construction implies univalent universes externally. An advantage of this formal approach is that our construction can also be interpreted in cubical sets on the connections cube category, and on the de Morgan cube category used in the CCHM model. As a first step towards comparing these approaches, we show that the two Kan operations are interderivable in a setting where both exist (presheaves on the de Morgan cube category, with the additional cofibration required by our construction).

1 Introduction

Cubical type theory is a formal system for Univalent Foundations/homotopy type theory [Voevodsky, 2006; The Univalent Foundations Program, Institute for Advanced Study, 2013]. The goals of cubical type theory are twofold: we would like a computational homotopy type theory, where terms can be interpreted as programs, and we would like a better syntax for synthetic homotopy theory.

1.1 Constructive cubical models and type theories

Bezem, Coquand, and Huber [2013] gave a constructive model of type theory in cubical sets. In this model, a syntactic type is interpreted as a cubical set equipped with a *Kan operation*, which generalizes the elimination rule for the identity type in Martin-Löf type theory. The standard Kan filling condition (given all but one face of a cube, there exists a missing face and the inside) is made algebraic (there is a chosen operation producing the missing face and inside), generalized to permit additional dimensions not involved in the filling problem, and made *uniform*, a naturality condition relating fillings along the maps in the cube category. The stability under reindexing expressed in this *uniform Kan condition* allows a constructive definition of the Kan operation for Π types, and is a generally better fit for type theory, where we expect strict stability under substitutions. The model includes path types defined using the interval object in cubical sets, which validate different judgemental equalities than the inductive identity types of Martin-Löf type theory. For example the identity type's computation rule (J on `refl`) is weakened to a path/propositional/typal equality, but many new equations hold exactly for the path type. An operational semantics based on this model was implemented in the `cubical` proof assistant prototype.[1]

Following the development of this model, efforts began to present cubical ideas in syntactic type theories, which has several benefits. For example, syntax for the intermediate states of the definitions of the Kan operations is required if one wants to apply judgemental equalities to reduce a complex (possibly open) expression, and then manually reason about the reduct—a very common step in type theoretic developments. Moreover, the path types and equalities in the cubical model provide a different language for synthetic homotopy theory, which is more convenient for at least some proofs than Martin-Löf type theory with univalence and higher inductive types as axioms. However, a central goal of synthetic homotopy theory is to do proofs in a model-independent way, which motivates investigating syntactic presentations of cubical ideas that could potentially be given more than one model. One approach to cubical type theory, explored by Altenkirch and Kaposi [2014]; Polonsky [2014], is to do an "internal inductive step" following Altenkirch et al. [2007]: define the identity type so that it computes to another type, so that higher-dimensional operations can be derived from lower-dimensional ones. This involves the fewest additions to type theory, but it is currently unknown whether it can be made to work for univalent universes. Another approach is to extend type theory with syntactic interval variables, which are interpreted using the interval object in cubical sets.

One degree of freedom in the design of constructive cubical models is the choice of cube category on which to take presheaves. Bezem, Coquand, and Huber [2013] use a cube category with faces, degeneracies, and symmetries. This cube category C is the free monoidal category $(C, \otimes, \cdot)$ generated by an object $\mathbb{I}$, with morphisms generated by face maps $0, 1 : \cdot \to \mathbb{I}$, degeneracy $\mathbb{I} \to \cdot$, and a symmetry involution $\mathbb{I} \otimes \mathbb{I} \to \mathbb{I} \otimes \mathbb{I}$ (which makes the category symmetric monoidal). Syntactically, this corresponds to substructural interval variables without contraction, as in nominal logic [Pitts, 2015]. In a syntactic type theory, it would be preferable to also have contraction, so that syntactic interval variables behave like ordinary variables, according to the standard structural rules. Semantically, contraction corresponds to diagonal maps in the cube category. In addition to simplifying the type theory, another motivation for diagonals, which was discovered during the implementation of the Bezem, Coquand, and Huber [2013] model, is that without them it is unclear whether one can give eliminators for higher inductive types that obey exact computation laws on path constructors. Adding diagonals to the cube category used in Bezem, Coquand, and Huber [2013] gives the *Cartesian cube category*, the free finite product category on an interval object (in the notation above, the $\otimes$ becomes a Cartesian product $\times$). Since 2013, Awodey has encouraged the investigation of the Cartesian cube category, and stressed its good mathematical properties, such as the fact that it is the classifying topos of bipointed sets, that (unlike the Bezem, Coquand, and Huber [2013] cube category) geometric realization to topological spaces preserves products, and that the interval is atomic (exponentiation by the interval has a right adjoint)—see Awodey [2016b]. The Cartesian cube category is a strict test category in the sense of Grothendieck, which means roughly that it has the same homotopy theory as simplicial sets/topological spaces; Buchholtz and Morehouse [2017] develop a thorough investigation of which cube categories are (strict)

[1] `https://github.com/simhu/cubical`

test categories.

Coquand developed a uniform Kan operation [Coquand, 2014b] for Cartesian cubical sets, and a uniform Kan operation [Coquand, 2014a] for cubical sets with diagonals plus additional degeneracies called *connections*, which can be thought of as meets and joins on the interval.[2] These two operations began to explore another degree of freedom in the design of constructive cubical models, which is what class of filling problems are allowed, and which composites of filling problems are taken as primitive. For example, the Kan operation designed for the Cartesian cube category allows more general trivial cofibrations, which are not necessary for the connections cube category because they can be encoded. Though there is certainly not a 1-1 correspondence between cube categories and Kan operations (for example, Awodey [2016a] considers the Cartesian cube category with a standard notion of trivial cofibration), for the remainder of this introduction, we will use "the Cartesian model" as shorthand for the Cartesian cube category with the [Coquand, 2014b] Kan operation, and "the connections model" as shorthand for the connections/de Morgan cube category with the Coquand [2014a] Kan operation.

Several syntactic type theories with interval variables were developed in mid-2014 in parallel independent work. Bernardy et al. [2015] developed a presheaf type theory for polymorphism. Coquand [2014a] developed a type theoretic presentation of the connections model. Brunerie and Licata [2014] developed a type theoretic presentation of the Cartesian model. Isaev [2014] developed an interval-based type theory with diagonals and (one) connection, and implemented a prototype proof assistant with path and higher inductive types using the interval; in this work, the interval itself is a "fibrant" type, and to our knowledge a fibrant interval has not yet been investigated from a semantic point of view.

Coquand [2014a] additionally introduced the "glue type" for extending a type by equivalences, a special case of which is univalence, and proposed a definition of a Kan composition structure on the universe. This connections model adopted a *regularity* condition, which is a generalization of the "`transport` on reflexivity is the identity" definitional equality in Martin-Löf type theory. With regularity, the path types of cubical type theory provide both the judgemental equalities of the MLTT identity type, and the useful new judgemental equalities that exponentiation by an interval provides. However, our attempts in 2014–2015 to adapt the glue types and definition of the universe from the connections model to the Cartesian model revealed that the proposed definition of the universe did not actually satisfy regularity.[3] Cohen, Coquand, Huber, and Mörtberg [2016] adapt the connections model to a non-regular setting, and show that the full univalence axiom follows from glue types. However, a solution for the Cartesian model with the more permissive Coquand [2014b] Kan operation remained elusive. Awodey [2016a] constructed a model of identity types in the Cartesian cube category using the standard notion of trivial cofibration (endpoint inclusions), where the identity types are interpreted as path types (exponentiation by the interval), and also satisfy regularity/normality, and therefore the full J on `refl` exact equality—but this has not yet extended to univalent universes.

In 2015 and 2016, both the connections model and the partial Cartesian model were studied from an operational point of view. Angiuli et al. [2016]; Angiuli and Harper [2017]; Angiuli et al. [2017a] developed a computational higher type theory based on Cartesian cubes. (Computational

[2]More precisely, the *de Morgan* cube category used in Coquand [2014a] also includes a reversal involution $1- : \mathbb{I} \to \mathbb{I}$, giving strict path reversal. We will gloss over the distinction between the connections cube category and the de Morgan one, as the differences between the connections and de Morgan Kan operations are less important to our story than the differences between the Cartesian and connections ones.

[3]See the post `https://goo.gl/btFxZ4` from 5/31/2015 on the Homotopy Type Theory mailing list.

type theory differs from the formal type theory considered here, in that an operational semantics is given first, and a logical relation on programs is taken as the definition of the type system.) Because the computational model validates the rules of the formal type theory, that work implies a canonicity result for Cartesian cubical type theory with Π, Σ, path types, booleans, the circle, and "isovalence" (gluing with a strict isomorphism, i.e., two functions that compose to the identity exactly, rather than up to paths). In parallel work, Huber [2016] proved canonicity for de Morgan cubical type theory (with connections and reversal), including Π, Σ, path, gluing (and therefore univalence), the universe, natural numbers, the circle, and propositional truncation.

In Spring 2017, Angiuli, Favonia, and Harper discovered how to construct univalent Kan universes for Cartesian cubes by adapting the [Coquand, 2014b] Kan operation. The key idea is to allow the diagonal inclusion $\mathbb{I} \to \mathbb{I} \times \mathbb{I}$ in the cube category as a cofibration, which geometrically corresponds to attaching faces to the diagonals of open boxes. Angiuli et al. [2017b] use these diagonal cofibrations to define a computational Cartesian cubical type theory with an infinite hierarchy of univalent universes as well as an extensional equality judgment internalized as an equality pretype.[4] In this paper, we consider diagonal cofibrations from a proof-theoretic and model-theoretic point of view, completing the Cartesian cubical type theory begun by Brunerie and Licata [2014], and the constructive model of univalence in Cartesian cubical sets begun by Coquand [2014b].

Though these two investigations use the same basic idea to construct the universe, Angiuli et al. [2017b]'s computational type theory is not a model of the formal type theory that we consider here. The differences are largely due to independent development, but they are also partly motivated by consideration of efficiency in evaluating programs. Efficiency is essential for using the computational content of proofs in practice—for example, we should be able to automatically reduce Brunerie's proof of $\pi_4(S^3)$ to determine that it is $\mathbb{Z}/2$, but this has not yet been possible in any implementation. Angiuli et al. [2017b] disallow false cofibrations, thereby avoiding the "empty system compositions" that proliferate in formalizations in the type theory of Cohen et al. [2016]. Additionally, univalence and Kan composition in the universe are treated separately: for univalence it suffices to consider only gluing with a single equivalence, and for composition one can avoid converting every path to an equivalence before composing. Moreover, Angiuli et al. [2017b] formulate a two-level type theory with hierarchies of both non-Kan pre-types and Kan types, including both exact equality and path types—these features may permit defining a fibrant type of semi-simplicial types. We encourage the reader to refer to that paper for these ideas, and here focus instead on the goal of finding minimal axioms that allow the universe to be Kan.

1.2 Kan Operations

The main aspects of our approach are best understood after a bit of background on cubical type theory and the syntax and semantics of uniform Kan operations. Our understanding of the semantics was particularly influenced by Awodey [2016a]; Gambino and Sattler [2017]; Orton and Pitts [2016]; Sattler [2017].

The basic idea of cubical type theory is to consider judgements of the form $\Psi; \Gamma \vdash a : A$, where Ψ is a context of dimension variables $x : \mathbb{I}$. So $\cdot; \Gamma \vdash a : A$ is a point, $x : \mathbb{I}; \Gamma \vdash a : A$ is a line, $x : \mathbb{I}, y : \mathbb{I}; \Gamma \vdash a : A$ is a square, and so on. Semantically, Ψ is an object of a cube category $\mathbb{C}$, which here we take to be the Cartesian cube category, the free finite product category on an interval object $\mathbb{I}$ with maps $\cdot \vdash 0 : \mathbb{I}$ and $\cdot \vdash 1 : \mathbb{I}$. A (closed) type is interpreted as a presheaf in $\hat{\mathbb{C}} := \mathbf{Sets}^{\mathbb{C}^{op}}$, and

[4] Some equality types are in fact Kan; see Angiuli et al. [2017b].

in the simple case where a, but not A, mentions variables from Ψ (and the context Γ is empty), the judgement $\Psi \vdash a : A$ represents a natural transformation from the representable $\hom_{\mathbb{C}}(-, \Psi)$ to A, or, by Yoneda, an element of $A(\Psi)$. Substitution of special symbols 0 and 1 for interval variables, which we write as $a\langle 0/x\rangle$ and $a\langle 1/x\rangle$, correspond to reindexing using the action of the presheaf A on the maps in the cube category. The structural rules of weakening, exchange, and contraction for interval variables correspond to reindexing by degeneracy, symmetry, and diagonals, respectively. The basic type constructors of type theory (Π, Σ, $\mathbb{N}$, etc.) can be interpreted in any presheaf model in such a way that their rules apply uniformly at every dimension. In a cubical type theory, this justifies stating their rules for an arbitrary context Ψ—e.g. for functions, we have both λ and application in every context Ψ. This generalization gives many constructions on paths, such as `ap` and function extensionality, as special cases of the rules for the basic type constructors.

To make this cubical set theory into a cubical (homotopy) type theory, we want types to be fibrations with respect to the paths given by maps from the dimension variables/interval object $\mathbb{I}$. This is accomplished by adding a *Kan filling operation*, whose basic shape is

$$\begin{tikzcd} \sqcup \arrow[r, "{[t,b]}"] \arrow[d] & \Gamma.A \arrow[d, two heads] \\ \square \arrow[r, "\theta"'] \arrow[ur, dashed, "\texttt{fill}"] & \Gamma \end{tikzcd} \qquad \begin{array}{c} \sqcup; \Gamma \vdash [t,b] : A \\ \hline \square; \Gamma \vdash \texttt{fill}_A(t,b) : A \\ \sqcup; \Gamma \vdash \texttt{fill}_A(t,b) \equiv [t,b] : A \end{array}$$

This says that given a fibration on the right, and a solvable filling problem on the left, a "whole" shape in Γ in the base, and a boundary in $\Gamma.A$ on the top, we obtain a whole shape in $\Gamma.A$. Commutativity of the outer square says that the boundary is in the same fiber in Γ as the whole shape in the base. Commutativity of the bottom triangle says that the filler is as well, and commutativity of the top triangle says that the filler agrees with the provided boundary on the base. The standard solvable filling problems introduced by Kan are the inclusions of a "box missing a face" into the whole box, e.g. an endpoint into a line, or three sides into a square. Here $\square$ is an object of the cube category/representable functor on it, and $\sqcup$ is a subpresheaf of it.

In syntax, this corresponds to the rule on the right. The base map θ does not appear in the syntax, but is implicit in the substitution principle for the judgement. The equation says that the top triangle commutes. More concretely, when the filling problem is actually the "left-bottom-right faces included into the square" suggested by the notation, we might write

$$\begin{array}{c} x : \mathbb{I}, y : \mathbb{I}; x = 0; \Gamma \vdash t_0 : A \qquad x : \mathbb{I}, y : \mathbb{I}; x = 1; \Gamma \vdash t_1 : A \qquad x : \mathbb{I}, y : \mathbb{I}; y = 0; \Gamma \vdash b : A \\ x : \mathbb{I}, y : \mathbb{I}; x = 0, y = 0; \Gamma \vdash t_0 \equiv b : A \qquad x : \mathbb{I}, y : \mathbb{I}; x = 1, y = 0; \Gamma \vdash t_1 \equiv b : A \\ \hline \begin{array}{c} x : \mathbb{I}, y : \mathbb{I}; (x = 0 \vee x = 1) \vee y = 0; \Gamma \vdash [[t_0, t_1], b] : A \\ \hline x : \mathbb{I}, y : \mathbb{I}; \Gamma \vdash \texttt{fill}_A(t,b) : A \\ x : \mathbb{I}, y : \mathbb{I}; x = 0; \Gamma \vdash \texttt{fill}_A(t,b) \equiv t_0 : A \\ x : \mathbb{I}, y : \mathbb{I}; x = 1; \Gamma \vdash \texttt{fill}_A(t,b) \equiv t_1 : A \\ x : \mathbb{I}, y : \mathbb{I}; y = 0; \Gamma \vdash \texttt{fill}_A(t,b) \equiv b : A \end{array} \end{array}$$

This says that we have a y-path t_0 at $x = 0$ (left), a y-path t_1 at $x = 1$ (right), and an x-path b at $y = 0$ (base), which agree on the corners, and we produce a whole square, which agrees with the provided sides on the boundary.

As this example illustrates, the boundaries of filling problems can be thought of as *formulas* or restrictions on the dimension context, an approach introduced by Cohen, Coquand, Huber, and Mörtberg [2016] and developed by Riehl and Shulman [2017]. The boundary formulas are often left-invertible, which means we could without loss of generality reduce the boundary to a sequence of terms. However, it is both convenient and more abstract to have syntax for boundaries. For example, we will define the filling operations in the type theory in such a way that they are *open-ended* with respect to extensions of the allowed filling problems, e.g. with the ones considered in Cohen, Coquand, Huber, and Mörtberg [2016]. Moreover, this approach is essential to presenting cubical models in the internal logic of a topos using techniques developed by Orton and Pitts [2016].

A key part of the specification of a cubical type theory/model is what filling problems are allowed. For example, it is necessary to preclude non-contractible shapes, fillers of which would make the path type inconsistent (for example, a line filler both of whose endpoints are specified would give a path between `true` and `false`). The standard open boxes considered by Kan have all but one face, which can be represented by a formula with $(x = 0 \vee x = 1)$ for every dimension *except* one, and one face in the remaining ("filling") direction. We call the former the "tube" and the latter the "cap" or "base".

However, cubical type theories represent dimensions as variables, and we generally want admissible and *silent* weakening (i.e., terms need not change when considered in a larger context), which is incompatible with requiring that open boxes have faces for all dimensions but one. Weakening corresponds to the action of degeneracy maps on the cubical sets, so for weakening to be admissible, the uniform Kan filling operations [Bezem et al., 2013] allow filling problems where some, but not necessarily all, interval variables are assigned faces in the tube, and impose equations relating filling problems before and after degeneracy.

We focus on explaining the generalized shapes of filling problems here; see [Awodey, 2016a; Gambino and Sattler, 2017] for an analysis of the equational aspects of uniformity in semantic terms. Suppose we have a boundary formula α on a context Ψ, which identifies the faces of Ψ that are part of the filling problem. For the standard open boxes, α will select both the 0 and 1 faces of all variables in Ψ, but it may select fewer. Then we can fill a $\Psi, z : \mathbb{I}$ cube if we are given (1) the tube sides specified by α, which all may depend on the filling direction z, and (2) a cap at $z = 0$, such that (3) these are compatible on both α and $z = 0$ (we could have an analogous rule for $z = 1$).

$$\begin{tikzcd}
(\Psi.\alpha, z : \mathbb{I}) \vee_{\Psi, z:\mathbb{I}} \Psi \arrow[r, "{[t,b]}"] \arrow[d, tail, "{[(\alpha\times z), 0/z]}"'] & \Gamma.A \arrow[d, two heads] \\
\Psi, z : \mathbb{I} \arrow[r, "\theta"] \arrow[ur, dashed, "\mathtt{fill}"] & \Gamma
\end{tikzcd}
\qquad
\frac{\begin{array}{c}\Psi, z : \mathbb{I} \vdash A\ \mathtt{Type} \\ \Psi.\alpha, z : \mathbb{I}; \Gamma \vdash t : A \\ \Psi; \Gamma \vdash b : A\langle 0/z\rangle \\ \Psi; \alpha; \Gamma \vdash t\langle 0/z\rangle \equiv b : A\langle 0/z\rangle\end{array}}{\begin{array}{c}\Psi, z : \mathbb{I}; \Gamma \vdash \mathtt{fill}_A(t, b) : A \\ \Psi, z : \mathbb{I}; \alpha; \Gamma \vdash \mathtt{fill}_A(t, b) \equiv t : A \\ \Psi; \Gamma \vdash (\mathtt{fill}_A(t, b))\langle 0/z\rangle \equiv b : A\end{array}}$$

The object in the upper-left corner is the pushout of the pullback of $\alpha \times z$ and $0/z$, which captures a tube and cap that fit together as above. That is, we begin with the boundary formula α, which determines a restricted object $\Psi.\alpha$ that includes into Ψ, and then take the pullback on the left, and then the pushout on the right:

$$\begin{tikzcd}
(\Psi.\alpha, z : \mathbb{I}) \times_{\Psi, z:\mathbb{I}} \Psi \arrow[r, hook, dashed, "f"] \arrow[d, hook, dashed, "s"] & \Psi.\alpha, z : \mathbb{I} \arrow[d, tail, "\alpha\times z"] \\
\Psi \arrow[r, tail, "0/z"] & \Psi, z : \mathbb{I}
\end{tikzcd}
\qquad
\begin{tikzcd}
(\Psi.\alpha, z : \mathbb{I}) \times_{\Psi, z:\mathbb{I}} \Psi \arrow[r, hook, "f"] \arrow[d, hook, "s"] & \Psi.\alpha, z : \mathbb{I} \arrow[d, hook, dashed] \\
\Psi \arrow[r, hook, dashed] & (\Psi.\alpha, z : \mathbb{I}) \vee_{\Psi, z:\mathbb{I}} \Psi
\end{tikzcd}$$

The pushout includes into $\Psi, z : \mathbb{I}$ by the universal property of the pushout applied to $\alpha \times z$ and $0/z$ (the pushout corner map).

The syntactic rule on the right expresses this pushout elimination principle: we need to give a tube t, which does depend on z but is restricted to α, along with a cap b, which is at $z = 0$, which agree on the pullback, i.e. on α and when $z = 0$. This form of the Kan operation is now closed under weakening/degenerating with additional interval variables, because we can inductively weaken all of A, α, t, b, and the boundary constraint, and then reapply the rule in the extended context. The fact that weakening "commutes" with the Kan operation is an aspect of *uniformity* [Bezem, Coquand, and Huber, 2013]; in full, uniformity says that the action of any cube map into Ψ commutes with the filling operation in the same way.

In model category terminology, the boundary formula α is a notion of *cofibration*. For Bezem, Coquand, and Huber [2013] filling problems, α is a disjunction of pairs $x = 0 \vee x = 1$ for variables x, capturing the fact that every dimension involved in the filling problem has both a 0 and 1 face, but some dimensions may not be involved. However, we could also take α to be disjunctions of $x = 0$ and $x = 1$ separately, in which case a dimension can be involved in the filling problem with only one specified tube side. Cohen, Coquand, Huber, and Mörtberg [2016] additionally allow conjunctions (pullbacks) of boundary formulas as cofibrations. The choice of cofibrations is quite flexible; for example, classical Cisinski model structures [Cisinski, 2006] take the cofibrations to be the much larger set of all monomorphisms (in this case, this would mean all monomorphisms in the presheaves, not just the monomorphisms in the cube category), though in a constructive setting, cofibrations need to be restricted to be decidable. Thus, an important degree of freedom in the design of a cubical type theory is what cofibrations are allowed. Next, $[(\alpha \times z), 0/z]$ (an arbitrary collection of faces crossed with an interval, and connected at $z = 0$) is a contractible subshape of Ψ, and is a *trivial cofibration*. The Kan operation says that lifting problems specified by a trivial cofibration can be solved in any type, because all types are intended to be *fibrations*.

Syntactically, a next problem to solve is that the above rule has a designated interval variable z in the rule's *conclusion*'s context, which means substitution will not be admissible in the standard way—we cannot think of the action of the cube maps as simply replacing interval variables with interval terms. The way to resolve this is to define a Kan *composition* operation instead of a Kan *filling* operation, which means that the operation produces only the "missing face" of an open box rather than the "missing face and inside". This is the Kan operation used in Cohen, Coquand, Huber, and Mörtberg [2016], with the de Morgan cube category, and can be drawn as follows (see Sattler [2017]):

$$\begin{tikzcd}
\Psi.\alpha \arrow[r, "\mathtt{inl}(id \times 1/z)"] \arrow[d, "\alpha"'] & (\Psi.\alpha, z : \mathbb{I}) \vee_{\Psi, z:\mathbb{I}} \Psi \arrow[r, "{[t,b]}"] \arrow[d, tail, "{[(\alpha \times z), 0/z]}"] & \Gamma.A \arrow[d, two heads] \\
\Psi \arrow[r, "id_\Psi \times 1/z"'] \arrow[urr, dashed] & \Psi, z : \mathbb{I} \arrow[r, "\theta"'] & \Gamma
\end{tikzcd}$$

$$\frac{\begin{array}{c} \Psi, z : \mathbb{I} \vdash A\,\mathtt{Type} \\ \Psi.\alpha, z : \mathbb{I}; \Gamma \vdash t : A \\ \Psi; \Gamma \vdash b : A\langle 0/z \rangle \\ \Psi.\alpha; \Gamma \vdash t\langle 0/z \rangle \equiv b : A\langle 0/z \rangle \end{array}}{\begin{array}{c} \Psi; \Gamma \vdash \mathtt{com}_{z.A}(\alpha \mapsto z.t)b : A\langle 1/z \rangle \\ \Psi.\alpha; \Gamma \vdash \mathtt{com}_{z.A}(\alpha \mapsto z.t)b \equiv t\langle 1/z \rangle : A\langle 1/z \rangle \end{array}}$$

Rather than getting a map from the middle-bottom, we get a map from the bottom-left, which is the "1-endpoint", or *composite* of the filling problem. Commutativity of the upper triangle says that at α, this restricts to the 1-endpoint of the tube. The reason that the restriction to composition is sufficient is that filling can in fact be derived from composition using connections.

However, Coquand's earlier Kan operation for diagonals [Coquand, 2014b] generalizes this composition operation (we present a simplified version first):

$$
\begin{tikzcd}
\Psi.\alpha \arrow[r, "\mathtt{inl}(id\times r'/z)"] \arrow[d, "\alpha"] & (\Psi.\alpha, z:\mathbb{I}) \vee_{\Psi,z:\mathbb{I}} \Psi \arrow[r, "{[t,b]}"] \arrow[d, tail, "{[(\alpha\times z),r/z]}"] & \Gamma.A \arrow[d, two heads] \\
\Psi \arrow[r, "id_\Psi \times r'/z"'] \arrow[urr, dashed] & \Psi, z:\mathbb{I} \arrow[r, "\theta"'] & \Gamma
\end{tikzcd}
\qquad
\frac{\begin{array}{l}\Psi \vdash r : \mathbb{I} \\ \Psi \vdash r' : \mathbb{I} \\ \Psi, z : \mathbb{I}; \Gamma \vdash A\,\mathtt{Type} \\ \Psi.\alpha, z : \mathbb{I}; \Gamma \vdash t : A \\ \Psi; \Gamma \vdash b : A\langle r/z\rangle \\ \Psi; \alpha; \Gamma \vdash t\langle r/z\rangle \equiv b : A\langle r/z\rangle\end{array}}{\begin{array}{c}\Psi; \Gamma \vdash \mathtt{com}_A^{z:r\to r'}\,(\alpha \mapsto z.t)\,(b) : A\langle r'/z\rangle \\ \Psi.\alpha; \Gamma \vdash \mathtt{com}_A^{z:r\to r'}\,(\alpha \mapsto z.t)\,(b) \equiv t\langle r'/z\rangle : A\langle r'/z\rangle\end{array}}
$$

The generalization is that the "source" and "target" of the composition problem are not restricted to 0 and 1, the endpoints of the interval, but can also be other maps $\Psi \to \mathbb{I}$ in the cube category; we write r and r' for arbitrary such maps. Thus, this Kan operation adopts a more permissive notion of both trivial cofibration (the middle map in the diagram) and allowed composite (the left-hand square in the diagram) than the Coquand [2014a] operation (with connections and reversal, the generalization is not necessary because the more general filling problems can be encoded). In the case of the Cartesian cube category, the additional trivial cofibrations/composites that are allowed by this Kan operation are product projections from the context (variables). We read the notation as "compose from r to r' in the z direction, with the tube t (which can also depend on z) on α, starting at r with b." Composing to or from a variable, as in $\mathtt{com}_A^{z:0\to x}\,(b)$ or $\mathtt{com}_A^{z:x\to 0}\,(b)$, can be thought of as "moving" an element b to or from a diagonal, because the source/target x can also occur in b. For example, consider the case where source is a variable x, the cap depends on x, and the tube is empty, $\mathtt{com}_A^{z:x\to 0}\,(b)$. Here, $x : \mathbb{I} \vdash b : A\langle x/z\rangle$ is a heterogeneous path in the line A, while $\mathtt{com}_A^{z:x\to 0}\,(b) : A\langle 0/x\rangle$ is a homogeneous path in the fiber over 0, whose endpoints are $\mathtt{com}_A^{z:1\to 0}\,(b)$ and b. So, this instance of composition is a *homogenization* operation that turns a heterogeneous path ("path over") into a homogeneous-path-with-a-`transport`. Dually, Kan filling should be $\mathtt{com}_A^{z:r\to z'}\,(\alpha \mapsto z.t)\,(b)$ for a fresh variable z'—filling is obtained by degenerating all components of the composition problem in a fresh direction, and then moving to the diagonal between the fresh direction and the original filling direction.

The full operation given by Coquand [2014b] includes an additional equality constraint, which is necessary for deriving filling and homogenization in this way: we need $\mathtt{com}_A^{z:r\to r}\,(\alpha \mapsto t)\,(b) \equiv b$. That is, moving from r to itself is the identity, whether r is 0, 1 or a variable. Thus, Coquand [2014b] requires that on $r = r'$, the composite is b, which we can notate as follows:

Definition 1 (Diagonal Kan composition [Coquand, 2014b]).

$$\begin{tikzcd}
(\Psi.\alpha) \vee_{\Psi} (\Psi.r = r') \arrow[r, "{[\mathtt{inl}(id \times r'/z), \mathtt{inr}(r=r')]}"] \arrow[d, "{[\alpha, r=r']}"] & (\Psi.\alpha, z : \mathbb{I}) \vee_{\Psi, z:\mathbb{I}} \Psi \arrow[r, "{[t,b]}"] \arrow[d, tail, "{[(\alpha \times z), r/z]}"] & \Gamma.A \arrow[d, two heads] \\
\Psi \arrow[r, "id_{\Psi} \times r'/z"'] \arrow[urr, dashed] & \Psi, z : \mathbb{I} \arrow[r, "\theta"'] & \Gamma
\end{tikzcd}$$

$$\frac{\begin{array}{c}\Psi \vdash r : \mathbb{I} \quad \Psi \vdash r' : \mathbb{I} \quad \Psi, z : \mathbb{I}; \Gamma \vdash A\,\mathtt{Type} \\ \Psi.\alpha, z : \mathbb{I}; \Gamma \vdash t : A \quad \Psi; \Gamma \vdash b : A\langle r/z\rangle \quad \Psi; \alpha; \Gamma \vdash t\langle r/z\rangle \equiv b : A\langle r/z\rangle\end{array}}{\begin{array}{c}\Psi; \Gamma \vdash \mathtt{com}_A^{z:r \to r'}\,(\alpha \mapsto z.t)\,(b) : A\langle r'/z\rangle \\ \Psi.\alpha; \Gamma \vdash \mathtt{com}_A^{z:r \to r'}\,(\alpha \mapsto z.t)\,(b) \equiv t\langle r'/z\rangle : A\langle r'/z\rangle \\ \Psi.r = r'; \Gamma \vdash \mathtt{com}_A^{z:r \to r'}\,t\,(b) \equiv b : A\langle r/z\rangle\end{array}}$$

The syntactic rule is the composition rule given in Brunerie and Licata [2014], updated to use boundary formulas; variations on this rule were also used in Angiuli et al. [2016]; Angiuli and Harper [2017]. The presentation of the composition principle using boundary formulas is inspired by the diagonal constraints of Angiuli et al. [2017b]: In stating the boundary, we have used a more general subobject/formula than above. Before, we had only used equations $x = 0$ and $x = 1$, but here, we also need a general equality $r =_{\mathbb{I}} r'$, which semantically is a subobject of Ψ determined by pulling back the diagonal map $\mathbb{I} \to \mathbb{I} \times \mathbb{I}$. The pushout-of-pullback in the top-left corner of the diagram encodes both the "restricts to t on α" and the "restricts to b on $r = r'$" constraints (and compatibility on both).

Coquand [2014b]; Brunerie and Licata [2014] give definitions of diagonal Kan composition (satisfying this $r = r'$ constraint) for Π, Σ, path, and some higher inductive base types. Angiuli et al. [2016]; Angiuli and Harper [2017] study variations on these definitions in an operational setting, and define this Kan operation for Π, Σ, path, higher inductive base types, and gluing with a strict isomorphism. However, this strict $r = r'$ constraint was a long-time obstacle to defining univalence/gluing with an equivalence, or showing that the universe itself is fibrant. For example, diagonal Kan composition can be encoded using connections and the $0 \to 1$ Kan composition, and when one does so, the $r = r'$ constraint becomes an instance of the regularity condition that proved problematic in that model.

Here, we show that it is possible to constructively define diagonal Kan composition for glue (equivalence extension) types, *if* the cofibrations α include the diagonal map $\mathbb{I} \to \mathbb{I} \times \mathbb{I}$ from the cube category, as in Angiuli et al. [2017b]. Viewing subobjects as propositions (maps into the subobject classifier of the topos), this corresponds to taking the proposition $r =_{\mathbb{I}} r'$, for any maps $r, r' : \Psi \to \mathbb{I}$ in the cube category, to be a cofibration. Geometrically, this corresponds to attaching faces on the diagonal of an open box—interestingly, like connections, this has somewhat of a simplicial flavor. In a classical setting, diagonal maps are cofibrations in, for example, Cisinski model structures, where the cofibrations are all monomorphisms. In a constructive setting, there is an obligation that cofibrations be decidable subobjects, but this is true for pullbacks of the diagonal $\mathbb{I} \to \mathbb{I} \times \mathbb{I}$ because equality of maps in the cube category is decidable. Fibrant glue types imply Kan and univalent universes, in the same manner as in Cohen, Coquand, Huber, and Mörtberg [2016].

Relative to the prior work by Awodey [2016a] on constructing models of type theory with the Cartesian cube category, our Kan operation differs by using a different class of trivial cofibrations (the source can be an arbitrary map in the cube category, not only an endpoint), by taking composition rather than filling as primitive (but for a general enough notion of composite that it includes filling), and by using a different class of cofibrations (allowing the diagonal $\mathbb{I} \to \mathbb{I} \times \mathbb{I}$). Our definition of the universe satisfies univalence but not regularity, and therefore our interval-based path type satisfies the J computation rule of Martin-Löf type theory only weakly (up to a path). Awodey's supports regularity/normality but not yet (to our knowledge) fibrant univalent universes. Thus, it remains open whether it is possible to combine the benefits of both approaches, and have a path type given as exponentiation by an interval, which both satisfies the J computation rule strictly and supports univalent and Kan universes.

1.3 Contributions

In Section 2, we give a formal cubical type theory based on the Cartesian cube category with Π, Σ, path, identity, natural number, boolean, suspension (we comment on the generalization to pushouts), glue, and universe types. For each type, we give a judgemental equality rule "defining" the composition structure on that type. Of course, these "definitions" are just postulated axioms of judgemental equality, so it is important to validate them.

Orton and Pitts [2016] have developed a technique for describing cubical models in the internal logic of a 1-topos, by postulating an interval object, cofibrations, and certain other operations. This approach can be mechanized: in principle, one should use an extensional type theory, but in a pinch we can use Agda with function extensionality and uniqueness of identity proofs as a substitute [Hofmann, 1995]. Orton and Pitts [2016] mechanize much of the connections model [Cohen, Coquand, Huber, and Mörtberg, 2016] in this style, though it does not seem possible to axiomatize a fibrant universe itself internally without some extension.

In Section 3, we describe progress towards a mechanized constructive model in cubical sets. We have formalized the definition of diagonal Kan composition for glue, Π, Σ, path, identity, natural number, boolean, and pushout types.[5] Our mechanization postulates the definitions of interval, Π, Σ, positive types, and (exact) equality in the internal logic—i.e. we obtain the formation, introduction, elimination, and $\beta\eta$ rules for these types from the metalanguage, Agda. Then we make certain postulates about an interval type and cofibrations. Relative to the axioms in [Orton and Pitts, 2016], we replace the axioms for connections on the interval with the axiom that the diagonal inclusion $\mathbb{I} \to \mathbb{I} \times \mathbb{I}$ in the cube category $\mathbb{C}$ is a cofibration (i.e. that the proposition $r =_{\mathbb{I}} r'$ in the internal logic is cofibrant, for arbitrary terms r, r' in the interval). We also use "propositional univalence" (interprovable cofibrations are equal) and the axiom that cofibrations are closed under conjunction only to construct identity types (where J on `refl` satisfies an exact equality) from path types using Swan's technique [Cohen, Coquand, Huber, and Mörtberg, 2016].

From these assumptions, our mechanization verifies all of the details of the construction of Kan composition operations for these types, e.g. checking that the constructions in Section 2.11 type check and have the correct boundaries. For presheaf toposes, the internal logic construction implies:

Theorem 1. *Let C be a finite product category with an object $\mathbb{I}$, with maps $0, 1 : 1 \to \mathbb{I}$ with $0 \neq 1$. In $\hat{C} := Sets^{C^{op}}$, suppose Cof is a subobject of Ω_{dec}, the subobject classifier for decidable propositions, which is closed under $=_{\mathbb{I}}$, $\vee$ and $\forall x : \mathbb{I}-$. For any cubical set Γ, write $FTy(\Gamma)$ for*

[5] `https://github.com/dlicata335/cart-cube`

semantic types (in some universe) over Γ *equipped with a diagonal Kan composition structure (Definition 1) for cofibrations classified by* Cof*. Then* $FTy(\Gamma)$ *is closed under semantic* Π*,* Σ*, path, and glue types. If* $\hat{C}$ *has the cubical sets corresponding to boolean, natural number, and pushout types, then* $FTy(\Gamma)$ *is closed under those as well. Finally, if* Cof *is closed under pullbacks, then* $FTy(\Gamma)$ *is closed under identity types as well.*

For example, we can take C to be the Cartesian cube category, with Cof anything from a minimal notion of cofibration, closed under just $\vee$ and $=_{\mathbb{I}}$ with $\forall$ defined by quantifier elimination, to a maximal one consisting of Ω_{dec} itself.

Fibrant glue types should imply univalent universes in the semantics as an immediate corollary, following [Cohen, Coquand, Huber, and Mörtberg, 2016], and using the definition suggested in Section 2.12.

We give a proof that the axioms used in our formalization are true in cubical sets on the Cartesian cube category, which is mostly the same as the argument in [Orton and Pitts, 2016], and sketch a proof that the fibrant glue types give univalent universes, which is mostly the same as the argument in [Cohen, Coquand, Huber, and Mörtberg, 2016]. We have not yet given an interpretation of the syntax in the model, but the definitions of the Kan operations in the two settings follow each other line by line.

The axioms are also true in cubical sets on the de Morgan cube category, which gives a model in the same category but using a different notion of cofibration than Cohen, Coquand, Huber, and Mörtberg [2016]. However, the two Kan conditions are interderivable if we have both *both* connections (and reversal) and diagonal cofibrations, which provides a potential route to comparing the approaches.

For reference, here is a summary of why each feature of our Kan operation, assumption about the interval, and closure condition for cofibrations is necessary for our definitions:

- Path types use the fact that there is an interval with endpoints 0 and 1.
- Composition for Π types use the fact that Kan filling is derivable from composition, and that the source and target of the composition operation can be interchanged. Therefore, this definition of Π types would not work without the generalized trivial cofibrations that we consider, because interchanging a filler results in a composite from a variable.
- Composition for Σ types use Kan filling.
- Composition for path types use the fact that endpoints $x = 0$ and $x = 1$ are cofibrations, and that $\vee$ is.
- Composition for strict base types (natural numbers, booleans) use the connectedness axiom.
- Composition for higher inductive types uses the reduction of composition to coercion and homogeneous composition, which in turn uses "homogenization", i.e. the generalized set of trivial cofibrations.
- Constructing glue types uses the strictification axiom, which in a constructive model requires decidable cofibrations.
- Composition for glue types uses closure of cofibrations under $\vee$, $=_{\mathbb{I}}$, and $\forall$.
- Composition for identity types uses closure of cofibrations under $\wedge$, and propositional univalence for cofibrations.

Acknowledgments. We thank Vladimir Voevodsky for his vision and leadership, creating a rich research area that has captivated all of us over the past 9 years and inspired the PhD theses of all four junior authors of this paper. We thank Jeremy Avigad, Steve Awodey, Evan Cavallo, Cyril Cohen, Fredrik Forsberg, Simon Huber, Ian Orton, Ed Morehouse, Anders Mörtberg, Andrew Pitts, Christian Sattler, Mike Shulman, Bas Spitters, and Jon Sterling for many helpful discussions and insights over the years leading to this work.

This material is based on research sponsored by The United States Air Force Research Laboratory under agreement number FA9550-15-1-0053 and and FA9550-16-1-0292. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the United States Air Force Research Laboratory, the U.S. Government, or Carnegie Mellon University.

2 Type Theory

2.1 Overview of Judgements

We have the following syntactic classes:

- $\Psi \vdash r : \mathbb{I}$ dimension terms / maps in the cube category
- $\Psi \vdash \phi\, \mathsf{boundary}$ subobjects
- $\Psi \vdash \alpha\, \mathsf{cofib}$ cofibrations
- $\phi \vdash_\Psi \alpha$ subobject ordering
- $\Psi; \phi \vdash \Gamma\ \mathsf{ctx}$ contexts
- $\Psi; \phi; \Gamma \vdash A\, \mathtt{Type}$ types
- $\Psi; \phi; \Gamma \vdash a : A$ terms

and judgemental equality

- $\Psi; \phi; \Gamma \vdash A \equiv A'\, \mathtt{Type}$
- $\Psi; \phi; \Gamma \vdash a \equiv a' : A$

We interleave discussion of the syntax and discussion of the "standard" model in presheaves on a cube category, though we plan to investigate whether other models are possible.

2.2 Contexts

We have three kinds of contexts: Ψ contains interval/dimension variables, ϕ contains cofibrations/face formulas, and Γ contains normal term variables. The dimension context Ψ is non-dependent; Ψ is in scope in the cofibration context ϕ, but ϕ is not internally dependent.

$$\begin{array}{rcl} \Psi & ::= & \cdot \mid \Psi, x : \mathbb{I} \\ \phi & ::= & \cdot \mid \phi, \alpha \\ \Gamma & ::= & \cdot \mid \Gamma, x : A \end{array}$$

The Ψ context has no formation conditions (aside from the usual invariant that variables are distinct, when implemented concretely), while ϕ and Γ require the context entries to be well-formed:

$$\frac{}{\Psi \vdash \cdot \;\mathsf{boundary}} \qquad \frac{\Psi \vdash \phi \;\mathsf{boundary} \quad \Psi \vdash \alpha \;\mathsf{cofib}}{\Psi \vdash \phi, \alpha \;\mathsf{boundary}} \qquad \frac{}{\Psi; \phi \vdash \cdot \;\mathsf{ctx}} \qquad \frac{\Psi; \phi \vdash \Gamma \;\mathsf{ctx} \quad \Psi; \phi; \Gamma \vdash A \;\mathsf{Type}}{\Psi; \phi \vdash \Gamma, x : A \;\mathsf{ctx}}$$

We overload the letter x for both term and dimension variables, because x is traditionally used for term variables, and writing dimension variables as x, y, z is extremely helpful for drawing pictures. When both are in play at once, we may write term variables as a, b, c.

2.3 Dimension terms

Dimension terms are 0, 1, and variables.

$$r ::= 0 \mid 1 \mid x$$

We write $\Psi \vdash r : \mathbb{I}$ to mean that r is either 0 or 1 or a variable from Ψ:

$$\frac{}{\Psi \vdash 0 : \mathbb{I}} \qquad \frac{}{\Psi \vdash 1 : \mathbb{I}} \qquad \frac{x : \mathbb{I} \in \Psi}{\Psi \vdash x : \mathbb{I}}$$

The dimension context behaves like a standard hypothetical judgement in all other judgements—unlike some of our previous attempts at cubical type theory, all rules treat dimension variables as placeholders, and do not, e.g. inspect whether a term is a variable, or whether two variables are different. Thus, we have "silent" weakening and exchange (we write a dotted line for a rule that is intended to be admissible):

$$\frac{\Psi, \Psi' \vdash J}{\Psi, x : \mathbb{I}, \Psi' \vdash J} \qquad \frac{\Psi, x' : \mathbb{I}, x : \mathbb{I}, \Psi' \vdash J}{\Psi, x : \mathbb{I}, x' : \mathbb{I}, \Psi' \vdash J}$$

and the substitution rule, with its usual composition law:

$$\frac{\Psi, x : \mathbb{I}, \Psi' \vdash J \quad \Psi \vdash r : \mathbb{I}}{\Psi, \Psi' \vdash J\langle r/x\rangle} \qquad \frac{}{J\langle r/x\rangle\langle r'/y\rangle \equiv J\langle r'/y\rangle\langle r\langle r'/y\rangle/x\rangle}$$

Substitution of dimension terms is defined in a completely standard way (by induction on syntax, replacing variables with terms).

Semantically, we think of a dimension context Ψ as an object in the cube category, or the representable presheaf on it. A dimension term $\Psi \vdash r : \mathbb{I}$ is a map into $\mathbb{I}$ in the cube category. Because all objects in the cube category are finite products of $\mathbb{I}$, we could define an n-place substitution judgement $\Psi \vdash \sigma : \Psi'$ representing all such maps as $|\Psi'|$-tuples of terms.

2.4 Cofibrations

We use a syntactic notion of cofibration, in the style introduced in Cohen, Coquand, Huber, and Mörtberg [2016]. Our presentation follows Riehl and Shulman [2017], and our discussion of their semantics follows Orton and Pitts [2016]; Sattler [2017].

Our type theory includes a notion of *boundary formula*, written $\Psi \vdash \phi\,\mathsf{boundary}$; later judgements will have the form $\Psi; \phi; \Gamma \vdash J$, where Ψ is a dimension context and ϕ is a boundary formula. A dimension context Ψ represents an object of the cube category $\mathbb{C}$, and therefore there is a representable presheaf $[\Psi] := \hom_{\mathbb{C}}(-, \Psi)$. A syntactic boundary formula $\Psi \vdash \phi\,\mathsf{boundary}$ can be thought of in several equivalent ways:

1. A map in $\hat{\mathbb{C}}$ (i.e. a natural transformation) from $[\Psi]$ to Ω, where Ω is the subobject classifier when $\hat{\mathbb{C}}$ is viewed as a topos.

2. A sieve on Ψ, i.e. a set of pairs $(\Psi', \hom_{\mathbb{C}}(\Psi', \Psi))$ that is closed under precomposition by arbitrary maps in $\mathbb{C}$. Since the subobject classifier in a presheaf topos sends an object Ψ to the set of sieves on Ψ, this is the same as (1) by Yoneda.

3. A subobject of $[\Psi]$, i.e. a subpresheaf of the representable on Ψ, i.e. another presheaf $\Psi.\phi$ with a monic natural transformation $\phi : \Psi.\phi \hookrightarrow [\Psi]$ that includes the subobject into the original. (Actually, a subobject is an isomorphism class of such monos, identifying (A, h) and (B, k) when there is an isomorphism between A and B that sends h to k.) That this is equivalent to (1) is the universal property of the subobject classifier.

   A map between subobjects ϕ_1 and ϕ_2 is a morphism $f : (\Psi.\phi_1 \to \Psi.\phi_2)$ that commutes with the inclusions into Ψ. Some standard facts about subobjects include: Any such commuting map is a monomorphism, because any map f such that $f; \phi_1 = \phi_2$ where ϕ_2 is mono is itself mono. Moreover, because ϕ_1 is mono, any two $f, g : \Psi.\phi_1 \to \Psi.\phi_2$ that satisfy $f; \phi_1 = \phi_2$ and $g; \phi_1 = \phi_2$ are equal—so we have a posetal subobject ordering. We write $\mathbf{Sub}(\Psi)$ for the category (poset) of subobjects.

Then *cofibrations* are a designated subset of boundary formulas (i.e. of maps into the subobject classifier, or of sieves, or of monos into $[\Psi]$), which used to generate the lifting problems solved by the Kan operation. We write $\mathbf{Cofibs}(\Psi)$ for the category of cofibrations into Ψ, where an object is a pair $(\Psi_0, \Psi_0 \rightarrowtail \Psi)$ such that the map is a mono that is in the designated set of cofibrations. Equivalently, we can think of a cofibration classifier that is itself a subobject of the subobject classifier, so it is a "subset of the subobjects". The syntactic judgement $\Psi \vdash \alpha\,\mathsf{cofib}$ is interpreted as a an object $(\Psi.\alpha, \alpha : \Psi.\alpha \hookrightarrow \Psi)$ in $\mathbf{Cofibs}(\Psi)$ or as a map into the cofibration classifier, and the rules for this judgement can be read as asserting that the cofibrations are closed under certain operations.

We now discuss the rules for forming boundaries and cofibrations.

Rules for Boundaries Syntactically, boundaries are represented by a context ϕ that is a list of cofibrations α. The empty boundary context is the identity map, while context extension ϕ, α means the pullback:

$$\begin{tikzcd}
\Psi.(\phi,\alpha) \arrow[r, hook, dashed] \arrow[d, tail, dashed] & \Psi.\alpha \arrow[d, tail] \\
\Psi.\phi \arrow[r, hook] & \Psi
\end{tikzcd}$$

with either of the equal maps $\Psi.(\phi, \alpha) \hookrightarrow \Psi$ (which are mono because they are the diagonal map of the pullback of two monos, so all four maps are monos). This pullback is the product $\phi \times \alpha$ in $\mathbf{Sub}(\Psi)$ (this holds in any topos)—regarding ϕ and α as maps into the subobject classifier, the pullback is their composition with $\wedge : \Omega \times \Omega \to \Omega$. This explains why the syntactic rules can treat ϕ, α as context extension.

In contrast with Cohen, Coquand, Huber, and Mörtberg [2016], it is not necessary to require these pullbacks/conjunctions to be cofibrations, except to construct identity types with an exact equality on `refl` (see Section 2.16).

Cofibrations We have the following cofibrations:

$$\alpha ::= (r = r') \mid \alpha_1 \vee \alpha_2 \mid \forall x.\alpha$$

which are well-formed as follows:

$$\frac{\Psi \vdash r : \mathbb{I} \quad \Psi \vdash r' : \mathbb{I}}{\Psi \vdash r = r'\ \mathsf{cofib}} \qquad \frac{\Psi \vdash \alpha_1\ \mathsf{cofib} \quad \Psi \vdash \alpha_2\ \mathsf{cofib}}{\Psi \vdash \alpha_1 \vee \alpha_2\ \mathsf{cofib}} \qquad \frac{\Psi, x : \mathbb{I} \vdash \alpha\ \mathsf{cofib}}{\Psi \vdash \forall x.\alpha\ \mathsf{cofib}}$$

The syntactic rules for the judgement $\Psi \vdash \alpha\ \mathsf{cofib}$ can be read as asserting that certain subobjects of Ψ are in the set of cofibrations. When viewed as maps into the subobject classifier, these subobjects are exactly the formulas indicated by the notation. The admissible substitution principle for $\Psi, x : \mathbb{I} \vdash \alpha\ \mathsf{cofib}$ says that cofibrations are closed under pullback along an arbitrary map in $\mathbb{C}$.

It is instructive to unpack these maps into the subobject classifier as monomorphisms. The cofibration $r = r'$ is (the Yoneda embedding of) a face, diagonal, or identity, or the unique map out of $\emptyset$ (the initial object of $\hat{\mathbb{C}}$), depending on the values of r and r'. Taking the diagonal inclusion / equality on the interval as a cofibration is the key ingredient used to define the diagonal Kan operation for the universe.

$$\begin{array}{ccccc}
\underline{x = 0 \text{ and } 0 = x} & \underline{x = 1 \text{ and } 1 = x} & \underline{x = y} & \underline{r = r} & \underline{0 = 1 \text{ and } 1 = 0} \\
\Psi & \Psi & \Psi, x : \mathbb{I} & \Psi & \emptyset \\
\big\downarrow{\scriptstyle id_\Psi \times 0/x} & \big\downarrow{\scriptstyle id_\Psi \times 1/x} & \big\downarrow{\scriptstyle id_\Psi \times x/y} & \big\downarrow{\scriptstyle id} & \big\downarrow{\scriptstyle !} \\
\Psi, x : \mathbb{I} & \Psi, x : \mathbb{I} & \Psi, x : \mathbb{I}, y : \mathbb{I} & \Psi & \Psi
\end{array}$$

Next, we consider the "or" $\alpha_1 \vee \alpha_2$. In $\mathbf{Sub}(\Psi)$, this is the coproduct $\alpha_1 + \alpha_2$, so the inference rule is asserting that the coproduct of two cofibrations is a cofibration. In terms of maps $\alpha_i : \Psi.\alpha_i \to \Psi$, this unpacks in any topos to the pushout of the pullback, which encodes an idea of *coherence* when α_1 and α_2 are both true:

$$\begin{tikzcd}
\Psi.(\alpha_1, \alpha_2) \arrow[r, hook, dashed, "f"] \arrow[d, hook, dashed, "s"] & \Psi.\alpha_1 \arrow[d, tail, "\alpha_1"] \\
\Psi.\alpha_2 \arrow[r, tail, "\alpha_2"] & \Psi
\end{tikzcd}
\qquad
\begin{tikzcd}
\Psi.(\alpha_1, \alpha_2) \arrow[r, hook, "f"] \arrow[d, hook, "s"] & \Psi.\alpha_1 \arrow[d, hook, dashed] \\
\Psi.\alpha_2 \arrow[r, hook, dashed] & \Psi.(\alpha_1 \vee \alpha_2)
\end{tikzcd}$$

with the cofibrations $\Psi.(\alpha_1 \vee \alpha_2) \rightarrowtail \Psi$ given by the universal property of the pushout applied to α_1 and α_2.

For $\forall x.\alpha$, we begin with $\Psi, x : \mathbb{I} \vdash \alpha\ \mathsf{cofib}$, i.e. a map $(\Psi, x : \mathbb{I}).\alpha \rightarrowtail (\Psi, x : \mathbb{I})$, i.e. an object of $\mathbf{Cofibs}(\Psi, x : \mathbb{I})$. There is a functor $- \times \mathbb{I} : \mathbf{Sub}(\Psi) \to \mathbf{Sub}(\Psi, x : \mathbb{I})$, which given a map

$\Psi.\beta \hookrightarrow \Psi$ produces $\beta \times 1 : \Psi.\beta, x : \mathbb{I} \hookrightarrow \Psi, x : \mathbb{I}$. Syntactically, this is weakening $\Psi \vdash \beta\,\mathsf{cofib}$ to $\Psi, x : \mathbb{I} \vdash \beta\,\mathsf{cofib}$. Semantically, this map is a cofibration, because it is a pullback of a cofibration along a map in $\mathbb{C}$. In any topos, the $\forall$ quantifier on subobjects is the right adjoint to pullback along the projection, $\forall : \mathbf{Sub}(\Psi, x : \mathbb{I}) \to \mathbf{Sub}(\Psi)$. So the syntactic rule $\Psi \vdash \forall x.\alpha\,\mathsf{cofib}$ is asserting that this right adjoint preserves cofibrations.

For most of our development, we do not need any *equations* between cofibrations, like $(\alpha \vee (0 = 1)) \equiv \alpha$ or $(\alpha_1 \vee \alpha_2) \equiv (\alpha_2 \vee \alpha_1)$, and require only that these cofibrations are interprovable. (These equations hold by "propositional univalence" of the subobject classifier in a topos.) We need these equations only when constructing identity types from path types (see Section 2.16).

Subboundaries Next, we have rules for the judgement $\phi \vdash_\Psi \alpha$, which means a map from ϕ to α in $\mathbf{Sub}(\Psi)$, i.e. a morphism $\Psi.\phi \to \Psi.\alpha$ that commutes over Ψ. Above, we remarked that this category is posetal; syntactically, we represent this by omitting proof terms for this judgement, so that terms that make use of different derivations of a sequent $\phi \vdash_\Psi \alpha$ are identified.

The rules are just the usual natural deduction rules for these connectives. The rules for $\vee$, for example, express the fact that $\alpha_1 \vee \alpha_2$ is a coproduct in $\mathbf{Sub}(\Psi)$, while the rules for context extension say that it is a product in $\mathbf{Sub}(\Psi)$.

$$\frac{\alpha \in \phi}{\phi \vdash_\Psi \alpha} \qquad \frac{\phi \vdash_\Psi \alpha_1}{\phi \vdash_\Psi \alpha_1 \vee \alpha_2} \qquad \frac{\phi \vdash_\Psi \alpha_2}{\phi \vdash_\Psi \alpha_1 \vee \alpha_2} \qquad \frac{\phi, \alpha_1 \vdash_\Psi \beta \quad \phi, \alpha_2 \vdash_\Psi \beta \quad \phi \vdash_\Psi \alpha_1 \vee \alpha_2}{\phi \vdash_\Psi \beta}$$

$$\frac{\phi \vdash_\Psi 0 = 1}{\phi \vdash_\Psi \beta} \qquad \frac{\Psi \vdash r : \mathbb{I}}{\phi \vdash_\Psi (r = r)} \qquad \frac{\begin{array}{l}\Psi, x : \mathbb{I} \vdash \phi'\,\mathsf{boundary} \\ \Psi, x : \mathbb{I} \vdash \alpha\,\mathsf{cofib} \\ \phi \vdash_\Psi r = r' \\ \phi, \phi'\langle r/x\rangle \vdash_\Psi \alpha\langle r/x\rangle\end{array}}{\phi, \phi'\langle r'/x\rangle \vdash_\Psi \alpha\langle r'/x\rangle}$$

$$\frac{\phi \vdash_{\Psi, x:\mathbb{I}} \alpha}{\phi \vdash_\Psi \forall x.\alpha} \qquad \frac{\phi \vdash_\Psi \forall x.\alpha \quad \Psi \vdash r : \mathbb{I}}{\phi \vdash_\Psi \alpha\langle r/x\rangle}$$

Boundaries and Types/Terms In type and term formation judgements, we use a context $\Psi; \phi; \Gamma \vdash J$. The $(\Psi; \phi)$ part can be understood as the domain $\Psi.\phi$ of the mono $\Psi.\phi \hookrightarrow [\Psi]$, or intuitively as "the subset of Ψ on which ϕ holds." All judgements should be compatible with admissible weakening/exchange/substitution principles:

$$\frac{\Psi; \phi, \phi' \vdash J}{\Psi; \phi, \alpha, \phi' \vdash J} \qquad \frac{\Psi; \phi, \alpha, \beta, \phi' \vdash J}{\Psi; \phi, \beta, \alpha, \phi' \vdash J} \qquad \frac{\Psi; \phi, \alpha \vdash J \quad \phi \vdash_\Psi \alpha}{\Psi; \phi \vdash J}$$

Most of our cofibrations are left-invertible, and we will need left rules towards other judgements, such as term and type formation. Here, we give rules for both, though if we identified types and elements of a universe, we could avoid the duplication.

When stating these and other rules, we use the following convention: We will often find it convenient to give a judgemental equality axiom concisely as "$\alpha \vdash t \equiv t'$". This should be understood as a rule that, in a general $\Psi; \phi; \Gamma$, with typing premises that make t and t' type check, *and an additional premise that* $\phi \vdash_\Psi \alpha$, the equation holds.

For contradiction we have

$$\frac{\phi \vdash_\Psi 0 = 1}{\Psi; \phi; \Gamma \vdash \texttt{abort} : A} \qquad 0 = 1 \vdash u \equiv \texttt{abort} \qquad \frac{\phi \vdash_\Psi 0 = 1}{\Psi; \phi; \Gamma \vdash \texttt{abort}\ \texttt{Type}} \qquad 0 = 1 \vdash A \equiv \texttt{abort}$$

For $\vee$, we use the pushout-of-pullback characterization, because $\Psi; \phi$ is the domain of $\Psi.\phi$ of the inclusion, which in this case is (below) a pushout of a pullback:

$$\frac{\phi \vdash_\Psi \alpha \vee \beta \quad \Psi; \phi, \alpha; \Gamma \vdash t : A \quad \Psi; \phi, \beta; \Gamma \vdash u : A \quad \Psi; \phi, \alpha, \beta; \Gamma \vdash t \equiv u : A}{\begin{array}{c} \Psi; \phi; \Gamma \vdash [\alpha \mapsto t, \beta \mapsto u] : A \\ \alpha \vdash [\alpha \mapsto t, \beta \mapsto u] \equiv t \\ \beta \vdash [\alpha \mapsto t, \beta \mapsto u] \equiv u \\ \alpha \vee \beta \vdash t \equiv [\alpha \mapsto t, \beta \mapsto t] \end{array}}$$

$$\frac{\phi \vdash_\Psi \alpha \vee \beta \quad \Psi; \phi, \alpha; \Gamma \vdash A\ \texttt{Type} \quad \Psi; \phi, \beta; \Gamma \vdash B\ \texttt{Type} \quad \Psi; \phi, \alpha, \beta; \Gamma \vdash A \equiv B\ \texttt{Type}}{\begin{array}{c} \Psi; \phi; \Gamma \vdash [\alpha \mapsto A, \beta \mapsto B]\ \texttt{Type} \\ \alpha \vdash [\alpha \mapsto A, \beta \mapsto B] \equiv A \\ \beta \vdash [\alpha \mapsto A, \beta \mapsto B] \equiv B \\ \alpha \vee \beta \vdash A \equiv [\alpha \mapsto A, \beta \mapsto A] \end{array}}$$

Additionally, we have congruence rules for $r = r'$:

$$\frac{\phi \vdash_\Psi r = r' \quad \Psi, x : \mathbb{I}; \phi, \phi'; \Gamma \vdash a : A}{\Psi; \phi, \phi\langle r'/x\rangle; \Gamma \vdash a\langle r/x\rangle \equiv a\langle r'/x\rangle : A\langle r/x\rangle} \qquad \frac{\phi \vdash_\Psi r = r' \quad \Psi, x : \mathbb{I}; \phi, \phi'; \Gamma \vdash A\ \texttt{Type}}{\Psi; \phi, \phi\langle r'/x\rangle; \Gamma \vdash A\langle r/x\rangle \equiv A\langle r'/x\rangle\ \texttt{Type}}$$

For example, from weakening and this, we can derive the following (we write an inference rule with a double-line to indicate a derivable (*not* invertible, as it is often used) rule):

$$\frac{\Psi \vdash r : \mathbb{I} \quad \Psi; \phi\langle r/x\rangle; \Gamma\langle r/x\rangle \vdash a : A\langle r/x\rangle}{\Psi, x : \mathbb{I}; \phi, (x = r); \Gamma \vdash a : A}$$

When we have a nested $\vee$ like $(\alpha_1 \vee \alpha_2 \vee \alpha_3 \vee \ldots \vee \alpha_n)$, we write $[\alpha_1 \mapsto t_1, \ldots, \alpha_n \mapsto t_n]$ for a nested case with the same associativity as the type itself.

2.5 Judgemental equality

All judgements respect equality of types and terms in all positions, e.g.

$$\frac{\Psi; \phi; \Gamma \vdash a : A \quad \Psi; \phi; \Gamma \vdash A \equiv A'\ \texttt{Type}}{\Psi; \phi; \Gamma \vdash a : A'}$$

We will often declare judgemental equality rules by simply writing $u \equiv v$; this should be taken to mean that the rule applies in all contexts, and has typing premises for each meta-variable appearing in the rule, which ensure that u and v are well-typed.

2.6 Term Structural Rules

Term structural rules are as usual:

$$\frac{x : A \in \Gamma}{\Psi; \phi; \Gamma \vdash x : A} \qquad \frac{\Psi; \phi; \Gamma, \Gamma' \vdash J}{\Psi; \phi; \Gamma, x : A, \Gamma' \vdash J}$$

$$\frac{\Psi; \phi; \Gamma, y : B, x : A, \Gamma' \vdash J}{\Psi; \phi; \Gamma, x : A, y : B, \Gamma' \vdash J} \qquad \frac{\Psi; \phi; \Gamma, x : A, \Gamma' \vdash J \quad \Psi; \phi; \Gamma \vdash a : A}{\Psi; \phi; \Gamma, \Gamma'[a/x] \vdash J[a/x]}$$

The composition laws should hold syntactically:

$$\begin{array}{rcll} u[a/x] & \equiv_\alpha & u & \text{when } x\#u \\ u[a/x][b/y] & \equiv_\alpha & u[b/y][a[b/y]/x] & \end{array}$$

We additionally have a composition law for term substitution and dimension substitution:

$$u[a/y]\langle r_0/x_0\rangle \equiv_\alpha u\langle r_0/x_0\rangle[a\langle r_0/x_0\rangle/y]$$

2.7 Kan Condition

Kan Composition All types come with a specified Kan operation:

$$\frac{\begin{array}{l} \Psi \vdash r, r' : \mathbb{I} \\ \Psi, z : \mathbb{I}; \phi; \Gamma \vdash A\ \mathtt{Type} \\ \Psi, z : \mathbb{I}; \phi, \alpha; \Gamma \vdash t : A \\ \Psi; \phi; \Gamma \vdash b : A\langle r/z\rangle \\ \Psi; \phi, \alpha; \Gamma \vdash t\langle r/z\rangle \equiv b : A\langle r/z\rangle \end{array}}{\begin{array}{c} \Psi; \phi; \Gamma \vdash \mathtt{com}_A^{z:r\to r'}\,(\alpha \mapsto z.t)\,(b) : A\langle r'/z\rangle \\ r = r' \vdash \mathtt{com}_A^{z:r\to r'}\,(\alpha \mapsto z.t)\,(b) \equiv b \\ \alpha \vdash \mathtt{com}_A^{z:r\to r'}\,(\alpha \mapsto z.t)\,(b) \equiv t\langle r'/z\rangle \end{array}}$$

The boundary conditions say that transporting from somewhere to itself (when r is equal to r') is the identity, and that on α, the result of the composition is the specified partial element at the destination r' (for example, the boundary of a composition of a "U" agrees with the top-left and top-right corners of the "U"). Congruence of judgemental equality for $\mathtt{com}_A^{z:r\to r'}\,(\alpha \mapsto z.t)\,(b)$ requires, in particular, that whenever two types are judgementally equal, they have the same Kan operation—this is the major challenge for glue types/univalence/the universe.

Filling from Composition Kan *filling*, as opposed to Kan *composition*, is the "whole square" instead of just the "missing side". We can derive filling by composing to a fresh variable, which can also be seen as degenerating and moving to a diagonal. For emphasis, we sometimes write

$$\frac{\text{same premises as } \mathtt{com} \quad z\#(r, A, \alpha, z'.t, b)}{\Psi; \phi; \Gamma \vdash \mathtt{fill}_A^{z':r\to z}\,(\alpha \mapsto z'.t)\,(b) := \mathtt{com}_A^{z':r\to z}\,(\alpha \mapsto z'.t)\,(b) : A\langle z/z'\rangle}$$

This has the following boundary:

$$\begin{array}{rcl} (\mathtt{fill}_A^{z':r\to z}\,(\alpha \mapsto z'.t)\,(b))\langle r/z\rangle & \equiv & b \\ (\mathtt{fill}_A^{z':r\to z}\,(\alpha \mapsto z'.t)\,(b))\langle r'/z\rangle & \equiv & \mathtt{com}_A^{z':r\to r'}\,(\alpha \mapsto z'.t)\,(b) \\ \alpha \vdash \mathtt{fill}_A^{z':r\to z}\,(\alpha \mapsto z'.t)\,(b) & \equiv & t[z \leftrightarrow z'] \end{array}$$

Homogeneous Composition and Coercion An *hcom structure* on a type A is a term as follows, satisfying the indicated boundary conditions:

$$\frac{\begin{array}{l}\Psi \vdash r, r' : \mathbb{I} \\ \Psi; \phi; \Gamma \vdash A : \mathtt{Type} \\ \Psi, z : \mathbb{I}; \phi, \alpha; \Gamma \vdash t : A \\ \Psi; \phi; \Gamma \vdash b : A \\ \Psi; \phi, \alpha; \Gamma \vdash t\langle r/z\rangle \equiv b : A\end{array}}{\begin{array}{c}\Psi; \phi; \Gamma \vdash \mathsf{hcom}^{r \to r'}_{A}\,(\alpha \mapsto z.t)\,(b) : A \\ r = r' \vdash \mathsf{hcom}^{r \to r'}_{A}\,(\alpha \mapsto z.t)\,(b) \equiv b \\ \alpha \vdash \mathsf{hcom}^{r \to r'}_{A}\,(\alpha \mapsto z.t)\,(b) \equiv t\langle r'/z\rangle\end{array}}$$

An hcom structure is like a composition structure, but the type A is not allowed to depend on the filling direction.

A *coercion structure* on a type $z.A$ is a term as follows

$$\frac{\begin{array}{l}\Psi \vdash r, r' : \mathbb{I} \\ \Psi, z : \mathbb{I}; \phi; \Gamma \vdash A : \mathtt{Type} \\ \Psi; \phi; \Gamma \vdash b : A\langle r/z\rangle\end{array}}{\begin{array}{c}\Psi; \phi; \Gamma \vdash \mathsf{coe}^{z:r \to r'}_{A}(b) : A\langle r'/z\rangle \\ r = r' \vdash \mathsf{coe}^{z:r \to r'}_{A}(b) \equiv b\end{array}}$$

An coercion structure is a composition structure with an empty “tube”/boundary constraint.

Every type has a homogeneous composition and coercion structure derived from composition:

$$\begin{array}{c}\mathtt{hcom}^{r \to r'}_{A}\,(\alpha \mapsto z.t)\,(b) := \mathtt{com}^{\_:r \to r'}_{A}\,(\alpha \mapsto z.t)\,(b) \\ \mathtt{coe}^{z:r \to r'}_{A}(b) := \mathtt{com}^{z:r \to r'}_{A}\,(0 = 1 \mapsto z.\mathtt{abort})\,(b)\end{array}$$

Conversely, if we have an hcom structure and a coercion structure, then we can define $\mathtt{com}$ in terms of these, i.e. it type- and boundary-checks to set

$$\mathsf{com}^{z:r \to r'}_{A}\,(\alpha \mapsto z.t)\,(b) := \mathsf{hcom}^{r \to r'}_{A\langle r'/z\rangle}\,((\alpha \mapsto z.\mathsf{coe}^{z:z \to r'}_{A}(t)))\,(\mathsf{coe}^{z:r \to r'}_{A}(b))$$

We do *not* take this as a judgemental equality in all types, but will make use of it in some types, when it is easier to define homogeneous composition and coercion separately. This was first (to our knowledge) used by Coquand [2015] in defining composition for higher inductive types. Various type theories have adopted this decomposition in every type, including Angiuli et al. [2017b], some of our earlier attempts at Cartesian cubical type theory, and an implementation of the connections model with regularity [Coquand, 2014a].

We use the sans-serif hcom for an arbitrary hcom structure, to emphasize that it is not necessarily judgementally equal to $\mathtt{hcom}$, the one defined in terms of the type's Kan operation. This is clearer in the semantics, where a type can exist separately from its Kan operation. In the syntax, every type has a specified Kan operation, but when giving judgemental equality rules “defining” it, we will sometimes want to e.g. first construct an hcom structure and a coercion structure, and then define it as above. But then, for example, the $\mathtt{hcom}$ defined from the type's $\mathtt{com}$ will not be judgementally equal to the original hcom structure that we used to define $\mathtt{com}$ in the first place, and the font difference is meant to (subtly) keep this distinction.

Strict Coercion from Homogeneous Composition and Weak Coercion For any type A, a weak coercion structure in A is a function (with the same typing rules as `coe`)

$$\mathsf{wcoe}_A^{z:r \to r'}\,(b) : A\langle r'/z\rangle$$

equipped with a path

$$\begin{array}{l} x; r = r' \vdash \beta_b : A \\ r = r' \vdash \beta_b\langle 0/x\rangle : \mathsf{wcoe}_A^{z:r\to r'}\,(b) \\ r = r' \vdash \beta_b\langle 1/x\rangle : b \end{array}$$

A key use of diagonal cofibrations is to make a strict coercion structure from a weak one, using homogeneous composition:

$$\mathsf{coe}_A^{w:s \to s'}(b) := \mathsf{hcom}_{A\langle s'/w\rangle}^{0 \to 1}\,(s = s' \mapsto x.\beta_b)\,(\mathsf{wcoe}_A^{w:s\to s'}\,(b))$$

Strictly Preserving Homogeneous Compositions Suppose we have a function $x : A \vdash f(x) : B(x)$, a homogeneous composition structure on A, and a composition structure on B. We say that *f strictly preserves hcoms if*

$$\begin{array}{ll} & f[(\mathsf{hcom}_A^{r\to r'}\,(\alpha \mapsto z.t)\,(b))/x] \\ \equiv & \mathsf{com}^{z:r\to r'}_{B[\mathsf{hcom}_A^{r\to z}\,(\alpha\mapsto z.t)\,(b)/x]}\,(\alpha \mapsto z.f[t/x])\,(f[b/x]) : C[(\mathsf{hcom}_A^{r\to r'}\,(\alpha \mapsto z.t)\,(b))/x] \end{array}$$

There is always a path between these, and in some types (HITs) it is a definitional equality.

2.8 Σ-types (as a negative type)

The introduction and elimination rules are as usual.

Formation, Intro, Elim

$$\frac{\Psi;\phi;\Gamma \vdash A\ \mathtt{Type} \qquad \Psi;\phi;\Gamma, x:A \vdash B\ \mathtt{Type}}{\Psi;\phi;\Gamma \vdash \Sigma x{:}A.B\ \mathtt{Type}} \qquad \frac{\Psi;\phi;\Gamma \vdash u : A \qquad \Psi;\phi;\Gamma \vdash v : B[u/x]}{\Psi;\phi;\Gamma \vdash (u,v) : \Sigma x{:}A.B}$$

$$\frac{\Psi;\phi;\Gamma \vdash u : \Sigma x{:}A.B}{\Psi;\phi;\Gamma \vdash \mathtt{fst}(u) : A} \qquad \frac{\Psi;\phi;\Gamma \vdash u : \Sigma x{:}A.B}{\Psi;\phi;\Gamma \vdash \mathtt{snd}(u) : B[\mathtt{fst}(u)/x]}$$

$$\begin{array}{rcl} \mathtt{fst}(u,v) & \equiv & u \\ \mathtt{snd}(u,v) & \equiv & v \\ u & \equiv & (\mathtt{fst}(u), \mathtt{snd}(u)) \end{array}$$

Kan Operation

$$\begin{array}{rcl} \mathtt{com}^{z:r\to r'}_{\Sigma x:A.B}\,(\alpha \mapsto z.t)\,(b) & \equiv & (\mathtt{com}_A^{z:r\to r'}\,(\alpha \mapsto z.\mathtt{fst}(t))\,(\mathtt{fst}(b)), \\ & & \mathtt{com}^{z:r\to r'}_{B[\mathtt{fill}_A^{z:r\to z}\,(\alpha\mapsto z.\mathtt{fst}(t))\,(\mathtt{fst}(b))/x]}\,(\alpha \mapsto z.\mathtt{snd}(t))\,(\mathtt{snd}(b))) \end{array}$$

The introduction, elimination, and β and η rules apply at any dimension; this means that we have β and η not only for pairing of points, but for pairing of paths, squares, etc. The computation rule for coercion/composition is a generalization of the usual rule for `transport` at Σ-type (push into both components, the second over the first).

2.9 Π-types

Formation, Intro, Elim

$$\frac{\Psi; \phi; \Gamma \vdash A\ \mathtt{Type} \qquad \Psi; \phi; \Gamma, x : A \vdash B\ \mathtt{Type}}{\Psi; \phi; \Gamma \vdash \Pi x{:}A.B\ \mathtt{Type}} \qquad \frac{\Psi; \phi; \Gamma \vdash f : \Pi x{:}A.B \qquad \Psi; \phi; \Gamma \vdash a : A}{\Psi; \phi; \Gamma \vdash f\ a : B[a/x]}$$

$$\frac{\Psi; \phi; \Gamma, x : A \vdash u : B}{\Psi; \phi; \Gamma \vdash \lambda x.u : \Pi x{:}A.B}$$

$$\begin{array}{rcl} (\lambda x.u)\ a & \equiv & u[a/x] \\ f & \equiv & \lambda\, x. f\ x \end{array}$$

Kan Operation

$$\mathtt{com}^{z:r \to r'}_{\Pi x:A.B}\,(\alpha \mapsto t)\,(b) \quad \equiv \quad \lambda a.\mathtt{com}^{z:r \to r'}_{B[\mathtt{fill}^{z:r' \to z}_{A}\ (a)/x]}\,(\alpha \mapsto z.t\,(\mathtt{fill}^{z:r' \to z}_{A}\ (a)/x))\,(b\,(\mathtt{coe}^{z:r' \to r}_{A}(a)))$$

In the introduction, we motivated Kan composition *to* arbitrary r', $\mathtt{com}^{z:0 \to r'}_{A}\,(\alpha \mapsto z.t)\,(b)$ (and analogously for 1), as a natural way to close the Kan filling operation under substitution. The "backward" coercion $\mathtt{coe}^{z:r' \to r}_{A}(a)$ and filling used here, swapping r and r', motivates Kan composition *from* arbitrary r.

2.10 Path Types

Formation, Introduction, Elimination

$$\frac{\Psi, x : \mathbb{I}; \phi; \Gamma \vdash A\ \mathtt{Type} \quad \Psi; \phi; \Gamma \vdash a_0 : A\langle 0/x \rangle \quad \Psi; \phi; \Gamma \vdash a_1 : A\langle 1/x \rangle}{\Psi; \phi; \Gamma \vdash \mathtt{Path}_{x.A}\,(a_0, a_1)\ \mathtt{Type}}$$

$$\frac{\Psi, x : \mathbb{I}; \phi; \Gamma \vdash u : A \quad \Psi; \phi; \Gamma \vdash u\langle 0/x \rangle \equiv a_0 : A \quad \Psi; \phi; \Gamma \vdash u\langle 1/x \rangle \equiv a_1 : A}{\Psi; \phi; \Gamma \vdash \Lambda x.u : \mathtt{Path}_{x.A}\,(a_0, a_1)}$$

$$\frac{\Psi; \phi; \Gamma \vdash u : \mathtt{Path}_{x.A}\,(a_0, a_1) \quad \Psi \vdash r : \mathbb{I}}{\Psi; \phi; \Gamma \vdash u\, r : A\langle r/x \rangle} \qquad u\,0 \equiv a_0 \qquad u\,1 \equiv a_1$$

$$\begin{array}{rcl} (\Lambda x.u)\ r & \equiv & u\langle r/x \rangle \\ u & \equiv & \Lambda s.(u\ s) \end{array}$$

These rules say that an element u of the path type can be turned back to a cube in A by instantiating it with a dimension term r. When r is a dimension variable that does not occur in u, this is simply choosing a name for the "hidden" dimension of the path type element. When r is a dimension variable that does occur, this is taking a diagonal. When r is 0 or 1, this is equal to the element specified by u's type, so the line connects what the type says it connects. The introduction rule inverts the elimination rules: to give an element of the path type, one must give a higher cube with the correct boundary.

Kan Operation

$$\begin{array}{rl} & \mathtt{com}^{z:r\rightarrow r'}_{\mathtt{Path}_{x.A}\,(x=r_i\mapsto a)}\,(\alpha\mapsto z.t)\,(b) \\ \equiv & \Lambda x.\mathtt{com}^{z:r\rightarrow r'}_{A}\,[\alpha\mapsto z.t\,x, (x=0)\mapsto \_.a_0, (x=1)\mapsto \_.a_1]\,(b\,x) \end{array}$$

The Kan operation extends the filling problem with the faces indicated by the type.

2.11 Glue Types

A glue type $\mathtt{Glue}\,(\alpha\mapsto(T,f))\,(B)$, introduced in Coquand [2014a]; Cohen et al. [2016], is a type that adjusts B by making it T on α, where f is a "partial function" $\alpha\vdash f:T\to B$. The formation, introduction, elimination, and $\beta\eta$ rules can be defined for `Glue` types for any function f—in the semantics, we do not insist that f is an equivalence at this stage. Moreover, our construction below shows that `Glue` types have a homogeneous composition structure for any function f. However, `Glue` types have coercion or a full composition structure only when f is an equivalence. The way we define the Kan operation for `Glue` below, the equivalence is needed only in one spot: showing that `Glue` has a *weak incoherent composition structure*. A weak incoherent composition structure could be defined differently for different definitions of equivalence—i.e., in the semantics, the same `Glue` cubical set could be shown to have (intensionally) different Kan operations for different definitions of equivalence. However, rather than considering non-Kan "pre-types" (bare cubical sets) in the type theory, we define `Glue` types only when f is an equivalence, for one particular notion of equivalence—in particular, we write $f : A\simeq B$ for a function $f:A\to B$ which is an hfiber-contractible equivalence (and we implicitly project f). ([Angiuli et al., 2017b] do consider non-Kan pre-types, but not `Glue` with non-equivalences.)

2.11.1 Formation, Introduction, Elimination

The type $\mathtt{Glue}\,(\alpha\mapsto(T,f))\,(B)$ is essentially a Σ type of $b:B$ and $\alpha\vdash t:T$ such that $f(t)\equiv b$; for example, the elimination rule is a projection to B. However, (a) we cannot say that boundary condition on its own in Kan types, and (b) it is stricter than such a Σ type because on α, the `Glue` type restricts to T, and the elements restrict to t, and the projection to B restricts to f.

$$\frac{\Psi;\phi;\Gamma\vdash B\ \mathtt{Type}\qquad \Psi;\phi,\alpha;\Gamma\vdash T\ \mathtt{Type}\qquad \Psi;\phi,\alpha;\Gamma\vdash f:T\simeq B}{\begin{array}{c}\Psi;\phi;\Gamma\vdash \mathtt{Glue}\,(\alpha\mapsto(T,f))\,(B)\ \mathtt{Type}\\ \alpha\vdash \mathtt{Glue}\,(\alpha\mapsto(T,f))\,(B)\equiv T\end{array}}$$

$$\frac{\begin{array}{l}\Psi;\phi;\Gamma\vdash b:B\\ \Psi;\phi,\alpha;\Gamma\vdash t:T\\ \Psi;\phi,\alpha;\Gamma\vdash f(t)\equiv b:B\end{array}}{\begin{array}{c}\Psi;\phi;\Gamma\vdash \mathtt{glue}\,(\alpha\mapsto t)\,(b):\mathtt{Glue}\,(\alpha\mapsto(T,f))\,(B)\\ \alpha\vdash \mathtt{glue}\,(\alpha\mapsto t)\,(b)\equiv t\end{array}}$$

$$\frac{\Psi;\phi;\Gamma\vdash a:\mathtt{Glue}\,(\alpha\mapsto(T,f))\,(B)}{\begin{array}{c}\Psi;\phi;\Gamma\vdash \mathtt{unglue}(a):B\\ \alpha\vdash \mathtt{unglue}(b)\equiv f(b)\end{array}}$$

$$\begin{array}{rcl}\mathtt{unglue}(\mathtt{glue}\,(\alpha\mapsto t)\,(b)) & \equiv & b\\ b & \equiv & \mathtt{glue}\,(\alpha\mapsto b)\,(\mathtt{unglue}(b))\end{array}$$

Defining composition for `Glue` is quite complex, so we break it into a few steps. Our presentation is chosen to isolate the use of individual features of the filling problems and cofibrations:

1. Glue types for any function have homogeneous composition. This uses only homogeneous composition in T and B, not coercion or full composition. This step uses a diagonal cofibration to validate the strict $r = r'$ constraint on homogeneous composition.

2. Glue types for an equivalence have a *weak incoherent coercion structure*. It is *weak* in the sense that we have a path, or "propositional β-reduction," relating $\mathsf{wcoe}^{z:r\rightarrow r'}_{\mathtt{Glue}\,(\alpha\mapsto(T,f))\,(B)}\,(b)$ and b when $r = r'$, rather than a definitional equality. It is *incoherent* in the sense that this operation does not necessarily restrict to $\mathsf{wcoe}^{z:r\rightarrow r'}_{T}\,(b)$ on $\forall z.\alpha$.

   This step is the only one that directly uses the assumption that f is an equivalence. We present an algorithm for hfiber-contractible equivalences, but one can change the definition of equivalence and the algorithm without affecting the rest of the construction. (It may be possible to define a type theory with non-fibrant types in which equivalences are defined as "weak coercion structures on gluing with f.")

3. Making an incoherent coercion from any weak incoherent coercion. This step works at any type, and is a key use of diagonal cofibrations.

4. Making an (incoherent) composition from (incoherent) coercion and homogeneous composition. This step also works at any type, and is the decomposition of `com` as `hcom` and `coe` discussed above—we reduce composition to a homogeneous composition by coercing the pieces into each fiber. This step uses "homogenization," or the fact that source of a composition problem can be a variable.

5. Making a coherent composition from an incoherent one by "aligning". In Cohen, Coquand, Huber, and Mörtberg [2016], aligning is "inlined" into the definition of the Kan structure for glue types; that it could be isolated as its own step was observed independently by Christian Sattler and Ian Orton (private communication). This is the only place where a $\forall x.\alpha$ cofibration is used.

2.11.2 `Glue` has homogeneous composition

First, we show that glue types have a homogeneous composition structure. This uses only homogeneous composition in T and B, not coercion or full composition; it also uses an $s = s'$ constraint.

Given the premises of an hcom structure, we want to derive a term

$$
\begin{aligned}
&h := \mathsf{hcom}^{s\rightarrow s'}_{\mathtt{Glue}\,[\alpha\mapsto z.(T,f)]\,(B)}\,(\beta\mapsto w.u)\,(v)\\
&s = s' \vdash h \equiv v \equiv \mathtt{glue}\,(\alpha\mapsto v)\,(\mathtt{unglue}(v))\\
&\beta \vdash h \equiv u\langle s'/w\rangle \equiv \mathtt{glue}\,(\alpha\mapsto u\langle s'/w\rangle)\,(\mathtt{unglue}(u\langle s'/w\rangle))
\end{aligned}
$$

The premises give

$$
\begin{aligned}
&v : \mathtt{Glue}\,[\alpha\mapsto z.(T,f)]\,(B)\\
&w;\beta \vdash u : \mathtt{Glue}\,[\alpha\mapsto z.(T,f)]\,(B)\\
&u\langle s/w\rangle \equiv v
\end{aligned}
$$

Our plan is to construct some terms b_1 and t and define

$$\begin{array}{l} \mathsf{hcom}^{s\to s'}_{\texttt{Glue}\,[\alpha\mapsto z.(T,f)]\,(B)}\,(\beta\mapsto w.u)\,(v) \equiv \texttt{glue}\,(\alpha\mapsto t)\,(b_1) \\ b_1 : B \\ \alpha \vdash t : T \\ \alpha \vdash f(t) \equiv b_1 : B \end{array}$$

First, we can homogeneously compose the appropriate pieces of the inputs in each of B and T:

$$\begin{array}{l} b_0 := \texttt{hcom}^{s\to s'}_B\,(\beta\mapsto w.\texttt{unglue}(u))\,(\texttt{unglue}(v)) : B \\ t := \alpha \vdash \texttt{hcom}^{s\to s'}_T\,(\beta\mapsto w.u)\,(v) : T \end{array}$$

However, we cannot finish the construction with $\texttt{glue}^{z:r\to r'}\,(\alpha\mapsto t)\,(b_0)$: for this to be well-formed, we would need $f(t)$ to be judgementally equal to b_0 on α, which is not the case. On α, b_0 is the composite of $f(u)$ and $f(v)$, whereas we need f applied to the composite of u and v. (In the language of Book HoTT, it is like the difference between `(ap f p) o (ap f q)` and `ap f (p o q)`.)

To fix this, we create a term b_1 that adjusts b_0 by the homotopy between these (this corresponds to `pres` in Cohen, Coquand, Huber, and Mörtberg [2016]). First, on α, we create a line q (in a fresh direction x) between these:

$$\begin{array}{rcl} x,\alpha \vdash q & := & \texttt{hcom}^{s\to s'}_B\,S\,(f(v)) : B \\ S & := & (0/x \mapsto w.\texttt{hcom}^{s\to w}_T\,(\beta\mapsto w.f(u))\,(f(v)), \\ & & 1/x \mapsto w.f(\texttt{hcom}^{s\to w}_T\,(\beta\mapsto u)\,(v)), \\ & & \beta\mapsto w.f(u)) \\ \alpha\vdash q\langle 0/x\rangle & \equiv & b_0 \equiv \texttt{hcom}^{s\to s'}_T\,(\beta\mapsto w.f(u))\,(f(v)) \\ \alpha\vdash q\langle 1/x\rangle & \equiv & f(t) \equiv f(\texttt{hcom}^{s\to s'}_T\,(\beta\mapsto w.u)\,(v)) \end{array}$$

Next we define b_1 by attaching q to b_0, and remembering on β and $s = s'$ the value of b_0, so that we preserve the boundary conditions needed overall:

$$\begin{array}{rl} S' := & (\alpha\mapsto x.q, \\ & \beta\mapsto \_.\texttt{unglue}(u\langle s'/w\rangle), \\ & s=s'\mapsto \_.\texttt{unglue}(v)) \\ b_1 := & \texttt{hcom}^{0\to 1}_B\,S'\,(b_0) \end{array}$$

Though this is not necessary for how it is used below, this definition is coherent, in that it restricts to composition in T on α:

$$\alpha\vdash \mathsf{hcom}^{s\to s'}_{\texttt{Glue}\,[\alpha\mapsto z.(T,f)]\,(B)}\,(\beta\mapsto w.u)\,(v)\equiv t\equiv \texttt{hcom}^{s\to s'}_T\,(\beta\mapsto w.u)\,(v)$$

2.11.3 `Glue` has weak incoherent coercion

Weak glue rules The $\texttt{glue}\,(\alpha\mapsto t)\,(b)$ term pairs together a term $b : B$ and $\alpha\vdash t : T$ such that $\alpha\vdash f(t)\equiv b$. If we weakened the judgemental equality to a path, then t together with this path would be an element of the homotopy fiber of f at b:

$$\texttt{hfiber}(f,b) := \Sigma t{:}T.\texttt{Path}_B\,(f(t),b)$$

Using homogeneous composition in B, we can derive the following weakened version of the glue rules, where the constructor takes a homotopy fiber as input:[6]

$$\frac{b : B \quad \alpha \vdash h : \mathtt{hfiber}(f, b)}{\overline{\mathtt{wglue}\,(\alpha \mapsto h)\,(b) : \mathtt{Glue}\,(\alpha \mapsto (T, f))\,(B)}}$$

$$\frac{g : \mathtt{Glue}\,(\alpha \mapsto (T, f))\,(B)}{\overline{\mathtt{wglue}_\eta : \mathtt{Path}_{\mathtt{Glue}\,(\alpha \mapsto (T,f))\,(B)}\,(\mathtt{wglue}\,(\alpha \mapsto (g, \mathtt{refl}))\,(\mathtt{unglue}(g)), g)}}$$

In the η rule, we have used the fact that $\alpha \vdash (g, \mathtt{refl}) : \mathtt{hfiber}(f, \mathtt{unglue}(g))$, which makes sense because of the equation that $\alpha \vdash f \equiv \mathtt{unglue}(-)$.

These are defined as follows. For $\mathtt{wglue}\,(\alpha \mapsto h)\,(b)$, we adjust b by the path coming from the hfiber, so that it has $f(t)$ on its boundary:

$$\mathtt{wglue}\,(\alpha \mapsto h)\,(b) := \mathtt{glue}\,(\alpha \mapsto \mathtt{fst}(h))\,(\mathtt{hcom}_B^{1 \to 0}\,(\alpha \mapsto x.\mathtt{snd}(h)\ x)\,(b))$$

The η is essentially adjusting by a `refl o refl = refl` path: expanding the definition, we need

$$\mathtt{wglue}_\eta : \mathtt{Path}\,(\mathtt{glue}\,(\alpha \mapsto g)\,(\mathtt{hcom}_B^{1 \to 0}\,(\alpha \mapsto \_.\mathtt{unglue}(g))\,(\mathtt{unglue}(g))), g)$$

which is given by

$$\Lambda x.\mathtt{glue}\,(\alpha \mapsto g)\,(\mathtt{hcom}_B^{1 \to x}\,(\alpha \mapsto \_.\mathtt{unglue}(g))\,(\mathtt{unglue}(g)))$$

Weak incoherent coercion for an equivalence Our first step in defining a coercion operation for glue is to define a *weak incoherent coercion structure*, which we write as $\mathtt{wcoe}^{z:r \to r'}_{\mathtt{Glue}\,(\alpha \mapsto (T,f))\,(B)}\,(b)$. Weakness means that on $r = r'$, it is not judgementally equal to b, but that we have a path relating them. Incoherence means that on $\forall z.\alpha$, it is not necessarily equal to $\mathtt{coe}^{z:r \to r'}_T(b)$, which is necessary for "confluence" with how the glue type itself "reduces" (semantically, syntactic types denote both a type and a Kan operation, so equal types need to denote equal Kan operations). We will fix both of these issues in later steps below.

More formally, given $g : (\mathtt{Glue}\,(\alpha \mapsto (T, f))\,(B))\langle r/z \rangle$, our goal is to define a term

$$g' := \mathtt{wcoe}^{w:r \to r'}_{\mathtt{Glue}\,(\alpha \mapsto (T,f))\,(B)}\,(g) : (\mathtt{Glue}\,(\alpha \mapsto (T, f))\,(B))\langle r'/z \rangle$$

equipped with a path

$$x; r = r' \vdash \beta_g : \mathtt{Path}_{(\mathtt{Glue}\,(\alpha \mapsto (T,f))\,(B))\langle r/z \rangle}\,(\mathtt{wcoe}^{w:r \to r'}_{\mathtt{Glue}\,(\alpha \mapsto (T,f))\,(B)}\,(g), g)$$

We can first make

$$b' := \mathtt{coe}_B^{z:r \to r'}(\mathtt{unglue}(g)) : B\langle r'/z \rangle$$

[6] We can also define a path

$$\mathtt{wglue}_\beta : \mathtt{Path}_B\,(\mathtt{unglue}(\mathtt{wglue}\,(\alpha \mapsto h)\,(b)), b)$$

Moreover, observe that

$$(\mathtt{wglue}\,(\alpha \mapsto h)\,(b), \mathtt{wglue}_\beta) : \mathtt{hfiber}(\mathtt{unglue}(-), b)$$

and that on α, $\mathtt{hfiber}(\mathtt{unglue}(-), b) \equiv \mathtt{hfiber}(f, b)$. Our construction gives $\alpha \vdash (\mathtt{wglue}\,(\alpha \mapsto h)\,(b), \mathtt{wglue}_\beta) \equiv h$, though this and the β path are not used below.

by coercing in the base. Because $f\langle r'/z\rangle$ is an equivalence, we can make something in the fiber over b':

$$t' := \alpha \vdash \mathtt{fst}(\mathtt{snd}(f\langle r'/z\rangle)(b')) : \mathtt{hfiber}(f\langle r'/z\rangle, b')$$

so we have

$$g' := \mathtt{wglue}\,(\alpha \mapsto t')\,(b') : (\mathtt{Glue}\,(\alpha \mapsto (T, f))\,(B))\langle r'/z\rangle$$

as desired.

What remains is to give a path between g' and g on $r = r'$. On $r = r'$ and α, g itself is in the fiber over $b' \equiv \mathtt{unglue}(g)$:

$$\alpha, r = r' \vdash (g, \mathtt{refl}) : \mathtt{hfiber}(f\langle r/z\rangle, b')$$

because $(\mathtt{Glue}\,(\alpha \mapsto (T, f))\,(B))\langle r/z\rangle \equiv T\langle r/z\rangle$ and $f\langle r/z\rangle(g) \equiv \mathtt{unglue}(g)$ and the $\mathtt{coe}$ in b' vanishes. So contractibility of fibers gives

$$\alpha, r = r' \vdash p : \mathtt{Path}_{\mathtt{hfiber}(f\langle r/z\rangle, b')}\,(t', (g, \mathtt{refl}))$$

and congruence gives

$$r = r' \vdash \Lambda x.\mathtt{wglue}\,(\alpha \mapsto p\,x)\,(b) : \mathtt{Path}\,(g', \mathtt{wglue}\,(\alpha \mapsto (g, \mathtt{refl}))\,(b'))$$

because $g' := \mathtt{wglue}\,(\alpha \mapsto t')\,(b')$. Recall that on $r = r'$, b' is $\mathtt{unglue}(g)$, and we also have

$$r = r' \vdash \mathtt{wglue}_\eta : \mathtt{Path}\,(\mathtt{wglue}\,(\alpha \mapsto (g, \mathtt{refl}))\,(\mathtt{unglue}(g)), g)$$

by the weak η defined above. Homogeneously composing (transitivity) these two paths, using the previously defined homogeneous composition in $\mathtt{Glue}\,(\alpha \mapsto (T, f))\,(B)$, gives a path

$$r = r' \vdash \mathtt{Path}\,(g', g)$$

as desired.

2.11.4 `Glue` has diagonal Kan composition

Finally, we show that from the weak incoherent coercion structure and homogeneous composition structure (both of which we've defined above), we can create a full (coherent) composition operation. This step works for any weak incoherent coercion structure and homogeneous composition structure (not necessarily coherent)—it does not depend on any other properties of the ones defined above.

Suppose we have any weak incoherent composition structure

$$\begin{array}{l}
\mathsf{wcoe}^{w:s\rightarrow s'}_{\mathtt{Glue}\,[\alpha\mapsto z.(T,f)]\,(B)}\,(b)\\
x; s = s' \vdash p : \mathtt{Glue}\,[\alpha \mapsto z.(T, f)]\,(B)\langle s/w\rangle\\
s = s' \vdash p\langle 0/x\rangle \equiv \mathsf{wcoe}^{w:s\rightarrow s'}_{\mathtt{Glue}\,[\alpha\mapsto z.(T,f)]\,(B)}\,(b)\\
s = s' \vdash p\langle 1/x\rangle \equiv b
\end{array}$$

and homogeneous composition structure

$$\mathsf{hcom}^{s\rightarrow s'}_{\mathtt{Glue}\,[\alpha\mapsto z.(T,f)]\,(B)}\,(\beta \mapsto t)\,(b)$$

Strict Incoherent Coercion We can define a strict (but still incoherent) coercion structure by[7]

$$\begin{array}{l} \mathsf{icoe}^{w:s\to s'}_{\mathtt{Glue}\,[\alpha\mapsto z.(T,f)]\,(B)}(b) := \mathsf{hcom}^{0\to 1}_{\mathtt{Glue}\,[\alpha\mapsto z.(T,f)]\,(B)\langle s'/w\rangle}\,(s=s'\mapsto x.p)\,(\mathsf{wcoe}^{w:s\to s'}_{\mathtt{Glue}\,[\alpha\mapsto z.(T,f)]\,(B)}\,(b)) \\ s=s'\vdash \mathsf{icoe}^{w:s\to s'}_{\mathtt{Glue}\,[\alpha\mapsto z.(T,f)]\,(B)}(b)\equiv b \end{array}$$

This step makes it clear where in our definition regularity/normality breaks. Let G be $w:\mathbb{I}\vdash \mathtt{Glue}\,[\alpha\mapsto z.(T,f)]\,(B)$. For regularity to hold overall for glue types, we would need for this step that $\mathsf{icoe}^{w:s\to s'}_G(b)$ is equal to b when w does not occur in the type G. But for this to be true, we would need $\mathsf{wcoe}^{w:s\to s'}_G\,(b)$ to be equal to b when w does not occur in G, and moreover that $\mathsf{wcoe}^{w:s\to s'}_G\,(b)$ is (judgementally) equal to b when $s=s'$, and that p is constant on $s=s'$ (and indeed, if this were true, we would not need to strictify weak compositions in the first place). For example, if $\mathsf{wcoe}^{w:s\to s'}_G\,(b)$ were regular (canceled when $w\#G$), but p were not constant (depended on x), then even if the hcom in the above definition of $\mathsf{icoe}^{w:s\to s'}_G(b)$ were regular, it would not cancel by regularity, because its tube would not be degenerate—so regularity of $\mathsf{wcoe}^{w:s\to s'}_G\,(b)$ would not imply regularity of $\mathsf{icoe}^{w:s\to s'}_G(b)$. However, it does not seem seem possible to define weak coercion for a general equivalence so that p is constant on $s=s'$, because the definition of p makes use of the composite-canceling homotopies of the equivalence.

On the other hand, if f is a strict isomorphism, then a definition of coercion following isovalence in [Angiuli and Harper, 2017] would cancel judgementally both when $w\#G$ and $s=s'$, assuming regularity holds for B and T. Since all other steps of composition for glue types are regularity-preserving, we could therefore have gluing with a strict isomorphism in a (non-Kan) universe where the Kan operation is regular. The definitions of the Kan operations for Π, Σ, path, and strict base types preserve regularity, so they could be included in this universe as well. If this universe is restricted to strict sets (types where any two lines are judgementally equal), so that every line determines a strict isomorphism, then this universe would itself be (non-regular) Kan in the next universe, with the Kan operation given by gluing.

Incoherent Composition As noted above, we can define a composition structure $\mathtt{com}$ from a coercion structure coe and a composition structure hcom. In this case, we use icoe and hcom, which gives an incoherent composition operation, in the sense that it does not restrict to T on $\forall w.\alpha$. Let G be $\mathtt{Glue}\,[\alpha\mapsto z.(T,f)]\,(B)$, and we define icom as:

$$\begin{array}{l} \mathsf{icom}^{w:s\to s'}_G\,[\beta\mapsto w.t]\,(b) := \mathtt{hcom}^{s\to s'}_{G\langle s'/w\rangle}\,(\beta\mapsto w.\mathsf{icoe}^{w:w\to s'}_G(t))\,(\mathsf{icoe}^{w:s\to s'}_G(b)) \\ s=s'\vdash \mathsf{icom}^{w:s\to s'}_G\,[\beta\mapsto w.t]\,(b)\equiv b \\ \beta\vdash \mathsf{icom}^{w:s\to s'}_G\,[\beta\mapsto w.t]\,(b)\equiv t\langle s'/w\rangle \end{array}$$

Aligning Finally, we use the "aligning" operation due (independently) to Christian Sattler and Ian Orton. This isolates the uses of $\forall w.\alpha$ in [Cohen et al., 2016] in one place.

We define a correct composition by

$$\begin{array}{rl} \mathtt{com}^{w:s\to s'}_{\mathtt{Glue}\,[\alpha\mapsto z.(T,f)]\,(B)}\,(\beta\mapsto w.t)\,(b) := & \mathsf{icom}^{w:s\to s'}_{\mathtt{Glue}\,[\alpha\mapsto z.(T,f)]\,(B)}\,S\,(b) \\ S := & [\beta\mapsto w.t, \\ & \forall w.\alpha\mapsto w.\mathtt{fill}^{w:s\to w}_T\,[\beta\mapsto w.t]\,(b)] \end{array}$$

[7] We noted above that this works for any type, but here we are using an incoherent weak coercion structure to get an incoherent strict one; this is not quite an instance of a lemma we can state in syntax, without distinguishing a type from its Kan operation.

These all agree with each other and with b, and the right-hand side satisfies

$$\forall w.\alpha \vdash \mathsf{icom}^{w:s \to s'}_{\mathtt{Glue}\,[\alpha \mapsto z.(T,f)]\,(B)}\, S\,(b) \equiv \mathtt{com}^{w:s \to s'}_{T}\,[\beta \mapsto t]\,(b)$$

This is necessary for "confluence" with the equation for the $\mathtt{Glue}\,[\alpha \mapsto z.(T,f)]\,(B)$ type on α.

2.12 Universe

Depending on technical details of the intended semantics, we could define universes a la Tarski or a la Russell; here, we elide the $El(-)$ for notational simplicity.

Assume we have a universe of fibrant types closed under glue types, with the composition structure defined above. For the universe to be a Kan type, we need a (homogeneous, since there are no free variables in the type $\mathtt{U}$) composition structure on it, which unpacks to the fact that whenever we have $B : \mathtt{U}$ and $z, \alpha \vdash T : \mathtt{U}$ (with $\alpha \vdash T\langle r/z\rangle \equiv B$), we need a Kan type

$$\begin{gathered}\mathtt{com}^{z:r \to r'}_{\mathtt{U}}\,(\alpha \mapsto z.T)\,(B)\\ \alpha \vdash \mathtt{com}^{z:r \to r'}_{\mathtt{U}}\,(\alpha \mapsto z.T)\,(B) \equiv T\langle r'/z\rangle\\ r = r' \vdash \mathtt{com}^{z:r \to r'}_{\mathtt{U}}\,(\alpha \mapsto z.T)\,(B) \equiv B\end{gathered}$$

A simple way to see that this exists is to convert lines to equivalences and use the fact that $\mathtt{Glue}$ types are Kan for equivalences. (To avoid this, we could define the elements of the type to be the elements of a $\mathtt{Glue}$ type as below, and then follow the above construction of composition from weak incoherent coercion, but replace the definition of weak incoherent coercion for hfiber-contractible equivalences with a direct construction for lines.)

The simple version is:

$$\mathtt{com}^{z:r \to r'}_{\mathtt{U}}\,(\alpha \mapsto z.T)\,(B) \equiv \mathtt{Glue}\,[\alpha \mapsto (T\langle r'/z\rangle, (\mathtt{coe}^{z:r' \to r}_{T}(-), e)), r = r' \mapsto (B, (\lambda x.x, e'))]\,(B)$$

Here we need

$$e : \mathtt{isequiv}(\mathtt{coe}^{z:r' \to r}_{T}(-)) \qquad e' : \mathtt{isequiv}(\lambda x.x) \qquad \alpha, r = r' \vdash e \equiv e'$$

e' is defined in the usual way (using contractibility of singletons, which follows from composition and filling), and then we can define e by transporting it:

$$e := \mathtt{coe}^{z:r' \to r}_{\mathtt{isequiv}(\mathtt{coe}^{z:r' \to z}_{T}(-))}(e')$$

so that it automatically agrees with e' on $r = r'$.

This type satisfies the desired boundary equations

$$\begin{aligned} r = r' \vdash \mathtt{Glue}\,[\alpha \mapsto (T\langle r'/z\rangle, \mathtt{coe}^{z:r' \to r}_{T}(-)), r = r' \mapsto (B, \lambda x.x)]\,(B) &\equiv B\\ \alpha \vdash \mathtt{Glue}\,[\alpha \mapsto (T\langle r'/z\rangle, \mathtt{coe}^{z:r' \to r}_{T}(-)), r = r' \mapsto (B, \lambda x.x)]\,(B) &\equiv T\langle r'/z\rangle\end{aligned}$$

Following Cohen, Coquand, Huber, and Mörtberg [2016], we should be able to prove the univalence axiom for a fibrant universe containing glue types. For example, an equivalence $f : A \simeq B$ is converted to a path in the universe as follows:

$$\mathtt{ua}(e) := \Lambda x.\mathtt{Glue}\,(x = 0 \mapsto (A, f), x = 1 \mapsto (B, (\lambda x.x, e)))\,(B) : \mathtt{Path}_{\mathtt{U}}\,(A, B)$$

Writing G for the glue type, this path has the correct endpoints because $x = 0 \vdash G \equiv A$ and $x = 1 \vdash G \equiv B$.

Both the universe and univalence use the fact that on α, $\mathtt{Glue}\,(\alpha \mapsto (T, f))\,(B)$ restricts to T. This equation between types forces the equation between Kan operations that requires the final aligning step in the construction of the Kan operation for glue types.

To obtain the full univalence axiom, an internal argument shows that it suffices to give a $\mathtt{Path}_{A \to B}\,(\mathtt{coe}^{z:0 \to 1}_{\mathtt{ua}(e)\,z}(-), f)$ (i.e. a $\mathtt{ua}$ with a β path can be “fixed” to satisfy the rest of the univalence axiom, using ideas from Egbert Rijke and Martín Escardó). We have not yet checked the details, but the β path should follow from reducing the definition of composition in glue types and some path algebra.

2.13 Strict Booleans

The intro and elim rules are as usual:

$$\frac{}{\Psi; \phi; \Gamma \vdash \mathtt{bool}\ \mathtt{Type}} \qquad \frac{}{\Psi; \phi; \Gamma \vdash \mathtt{true} : \mathtt{bool}} \qquad \frac{}{\Psi; \phi; \Gamma \vdash \mathtt{false} : \mathtt{bool}}$$

$$\frac{\Psi; \phi; \Gamma, x : \mathtt{bool} \vdash C\ \mathtt{Type} \quad \Psi; \phi; \Gamma \vdash u : \mathtt{bool} \quad \Psi; \phi; \Gamma \vdash v_1 : C[\mathtt{true}/x] \quad \Psi; \phi; \Gamma \vdash v_2 : C[\mathtt{false}/x]}{\Psi; \phi; \Gamma \vdash \mathtt{if}_{x.C}(u, v_1, v_2) : C[u/x]}$$

$$\mathtt{if}_{x.C}(\mathtt{true}, v_1, v_2) \equiv v_1 \qquad \mathtt{if}_{x.C}(\mathtt{false}, v_1, v_2) \equiv v_2$$

The semantics in cubical sets (in particular the connectedness of the interval) justifies equality reflection for booleans:

$$\frac{\Psi, x : \mathbb{I}; \phi; \Gamma \vdash u : \mathtt{bool} \quad \Psi \vdash r, r' : \mathbb{I}}{\Psi; \phi; \Gamma \vdash t\langle r/x\rangle \equiv t\langle r'/x\rangle : \mathtt{bool}}$$

This rule implies the usual formulation of equality reflection: given a $\mathtt{Path}_{\mathtt{bool}}\,(b_0, b_1)$, $b_0 \equiv b_1$ (taking r, r' to be $0, 1$) and the path is reflexivity on b_0 say (taking r, r' to be $0, x$, where x is the Λ-bound variable of the path). In the presence of this rule, it type- and boundary-checks to define

$$\mathtt{com}^{\_:r \to r'}_{\mathtt{bool}}\,(\alpha \mapsto t)\,(b) \equiv b$$

The same definition works for any closed base type with decidable equality.

A more intensional presentation could be chosen instead to make type checking algorithmic, following Cohen, Coquand, Huber, and Mörtberg [2016]—only apply this equation when both t and b are equal to $\mathtt{true}$ or both are equal to $\mathtt{false}$.

Booleans could also be treated like a higher inductive type, where homogeneous compositions are values. For the notion of Kan operation that we use here, such booleans have non-standard points in the empty context (homogeneous compositions with a false face formula). Angiuli et al. [2017b] use a different definition of composition that disallows false cofibrations; as a result, the homotopy booleans have only $\mathtt{true}$ and $\mathtt{false}$ as points in the empty context.

2.14 Strict Natural Numbers

The intro and elim rules are as usual:

$$\frac{}{\Psi;\phi;\Gamma\vdash \texttt{nat}\ \texttt{Type}} \qquad \frac{}{\Psi;\phi;\Gamma\vdash \texttt{zero} : \texttt{nat}} \qquad \frac{\Psi;\phi;\Gamma\vdash u : \texttt{nat}}{\Psi;\phi;\Gamma\vdash \texttt{succ}(u) : \texttt{nat}}$$

$$\frac{\begin{array}{l}\Psi;\phi;\Gamma, x : \texttt{nat}\vdash C\ \texttt{Type}\\ \Psi;\phi;\Gamma\vdash u : \texttt{nat}\\ \Psi;\phi;\Gamma\vdash v_0 : C[\texttt{zero}/x]\\ \Psi;\phi;\Gamma, n : \texttt{nat}, r : C[n/x]\vdash v_1 : C[\texttt{succ}(n)/x]\end{array}}{\Psi;\phi;\Gamma\vdash \mathbb{N}{-}\texttt{elim}_{x.C}(u, v_0, x.r.v_1) : C[u/x]}$$

$$\mathbb{N}{-}\texttt{elim}_{x.C}(\texttt{zero}, v_0, v_1) \equiv v_1 \qquad \mathbb{N}{-}\texttt{elim}_{x.C}(\texttt{succ}(n), v_0, x.r.v_1) \equiv v_1\langle n/x\rangle\langle \mathbb{N}{-}\texttt{elim}_{x.C}(n, v_0, x.r.v_1)/r\rangle$$

Analogously to booleans, one possibility is to have equality reflection:

$$\frac{\Psi, x : \mathbb{I};\phi;\Gamma\vdash u : \texttt{nat} \quad \Psi\vdash r, r' : \mathbb{I}}{\Psi;\phi;\Gamma\vdash t\langle r/x\rangle \equiv t\langle r'/x\rangle : \texttt{bool}}$$

and

$$\texttt{com}_{\texttt{nat}}^{-:r\to r'}\,(\alpha\mapsto t)\,(b) \equiv b$$

2.15 $\mathbf{\Sigma} A$

We consider suspensions as an example higher inductive type; suspensions and booleans suffice to construct all spheres. The introduction forms are the usual HIT constructors, plus a homogeneous composition structure:

$$\frac{\Psi;\phi;\Gamma\vdash A\ \texttt{Type}}{\Psi;\phi;\Gamma\vdash \mathbf{\Sigma}A\ \texttt{Type}} \qquad \frac{}{\Psi;\phi;\Gamma\vdash \texttt{north} : \mathbf{\Sigma}A} \qquad \frac{}{\Psi;\phi;\Gamma\vdash \texttt{south} : \mathbf{\Sigma}A}$$

$$\frac{\Psi\vdash r : \mathbb{I} \quad \Psi;\phi;\Gamma\vdash u : A}{\Psi;\phi;\Gamma\vdash \texttt{merid}_r(u) : \mathbf{\Sigma}A} \qquad \texttt{merid}_0(u) \equiv \texttt{north} \qquad \texttt{merid}_1(u) \equiv \texttt{south}$$

$$\frac{\begin{array}{l}\Psi\vdash r, r' : \mathbb{I}\\ \Psi;\phi;\Gamma\vdash \mathbf{\Sigma}A : \texttt{Type}\\ \Psi, z : \mathbb{I};\phi,\alpha;\Gamma\vdash t : \mathbf{\Sigma}A\\ \Psi;\phi;\Gamma\vdash b : \mathbf{\Sigma}A\\ \Psi;\phi,\alpha;\Gamma\vdash t\langle r/z\rangle \equiv b : \mathbf{\Sigma}A\end{array}}{\begin{array}{c}\Psi;\phi;\Gamma\vdash \texttt{vhcom}_{\mathbf{\Sigma}A}^{r\to r'}\,(\alpha\mapsto t)\,(b) : \mathbf{\Sigma}A\\ r = r'\vdash \texttt{vhcom}_{\mathbf{\Sigma}A}^{r\to r'}\,(\alpha\mapsto z.t)\,(b) \equiv b\\ \alpha\vdash \texttt{vhcom}_{\mathbf{\Sigma}A}^{r\to r'}\,(\alpha\mapsto z.t)\,(b) \equiv t\langle r'/z\rangle\end{array}}$$

The eliminator computes on all of these:

$$\frac{\begin{array}{c}\Psi; \phi; \Gamma, a : \mathbf{\Sigma} A \vdash C\ \mathtt{Type} \\ \Psi; \phi; \Gamma \vdash u_0 : C[\mathtt{north}/a] \\ \Psi; \phi; \Gamma \vdash u_1 : C[\mathtt{south}/a] \\ \Psi, z : \mathbb{I}; \Phi; \Gamma, b : A \vdash u_2 : C[\mathtt{merid}_z(b)/a] \\ \Psi; \Phi; \Gamma, b : A \vdash u_2\langle 0/z\rangle \equiv u_0 : C[\mathtt{north}/a] \\ \Psi; \Phi; \Gamma, b : A \vdash u_2\langle 1/z\rangle \equiv u_1 : C[\mathtt{south}/a]\end{array}}{\Psi; \phi; \Gamma \vdash \mathbf{\Sigma}^{a.C}_{\mathrm{elim}}(u_0; u_1; z.b.u_2; u) : A[u/x]}$$

$$\begin{array}{rcl}\mathbf{\Sigma}^{a.C}_{\mathrm{elim}}(u_0; u_1; z.b.u_2; \mathtt{north}) & \equiv & u_0 \\ \mathbf{\Sigma}^{a.C}_{\mathrm{elim}}(u_0; u_1; z.b.u_2; \mathtt{south}) & \equiv & u_1 \\ \mathbf{\Sigma}^{a.C}_{\mathrm{elim}}(u_0; u_1; z.b.u_2; \mathtt{merid}_r(u')) & \equiv & u\langle r/z\rangle[u'/b] \\ \mathbf{\Sigma}^{a.C}_{\mathrm{elim}}(u_0; u_1; z.b.u_2; \mathtt{vhcom}^{r \to r'}_{\mathbf{\Sigma} A}\,(\alpha \mapsto t)\,(B)) & \equiv & \mathtt{com}^{z:r \to r'}_{C[\mathtt{vhcom}^{r \to z}_{\mathbf{\Sigma} A}\,(\alpha \mapsto z.t)\,(b)/x]}\,(\alpha \mapsto z.E(t))\,(E(b))\end{array}$$

In the final equation, we write $E(x)$ for $\mathbf{\Sigma}^{a.C}_{\mathrm{elim}}(u_0; u_1; z.b.u_2; x)$; the equation says that E strictly preserves hcoms, for $\mathtt{vhcom}^{r \to r'}_{\mathbf{\Sigma} A}\,(\alpha \mapsto t)\,(B)$ in the domain and the canonical composition structure on C in the range.

In the semantics, for each type A, we first make a cubical set satisfying the above rules, where the elimination rule can be used towards any type that is fibrant over $\mathbf{\Sigma} A$, following Coquand [2015] (i.e. there is a composition structure for $x : \mathbf{\Sigma} A \vdash C\ \mathtt{Type}$ as an additional argument, which in syntax is always the canonical composition structure for C). Next, we show that this cubical set has a Kan operation, and to do so we use a couple of instances of the elimination rule that are *not* for the canonical composition structure denoted by the type C. Thus, in syntax, we cannot write these as instances of the elimination rule (which fixes the translation of $\mathtt{vhcom}^{r \to r'}_{\mathbf{\Sigma} A}\,(\alpha \mapsto t)\,(B)$ as the canonical one for C), so we instead give them their own pattern-matching definitions.

We define a composition structure on $\mathbf{\Sigma} A$ by decomposing composition as homogeneous composition and coercion, and coercion as weak coercion and homogeneous composition. We use $\mathtt{vhcom}^{r \to r'}_{\mathbf{\Sigma} A}\,(\alpha \mapsto t)\,(b)$ as the homogeneous composition structure, so it suffices to define a weak coercion structure. First, we add the function (note that this is a new derivable rule of the type theory, which is an additional elimination principle for $\mathbf{\Sigma} A$):

$$\frac{\Psi \vdash r, r' : \mathbb{I} \qquad \Psi, z : \mathbb{I}; \phi; \Gamma \vdash A : \mathtt{Type} \qquad \Psi; \phi; \Gamma \vdash b : \mathbf{\Sigma}(A\langle r/z\rangle)}{\Psi; \phi; \Gamma \vdash \mathtt{wcoe}^{z:r \to r'}_{\mathbf{\Sigma} A}\,(b) : \mathbf{\Sigma}(A\langle r'/z\rangle)}$$

$$\begin{array}{rcl}\mathtt{wcoe}^{z:r \to r'}_{\mathbf{\Sigma} A}\,(\mathtt{north}) & \equiv & \mathtt{north} \\ \mathtt{wcoe}^{z:r \to r'}_{\mathbf{\Sigma} A}\,(\mathtt{south}) & \equiv & \mathtt{south} \\ \mathtt{wcoe}^{z:r \to r'}_{\mathbf{\Sigma} A}\,(\mathtt{merid}_{r_0}(u)) & \equiv & \mathtt{merid}_{r_0}(\mathtt{coe}^{z:r \to r'}_{A}\,(u)) \\ \mathtt{wcoe}^{z:r \to r'}_{\mathbf{\Sigma} A}\,(\mathtt{vhcom}^{r \to r'}_{\mathbf{\Sigma} A}\,(\alpha \mapsto t)\,(b)) & \equiv & \mathtt{vhcom}^{r \to r'}_{\mathbf{\Sigma} A}\,(\alpha \mapsto z.\mathtt{wcoe}^{z:r \to r'}_{\mathbf{\Sigma} A}\,(t))\,(\mathtt{wcoe}^{z:r \to r'}_{\mathbf{\Sigma} A}\,(b))\end{array}$$

The last line says that $\mathtt{wcoe}^{z:r \to r'}_{\mathbf{\Sigma} A}\,(-)$ strictly preserves hcoms, where the hcom structure for the input is $\mathtt{vhcom}^{r \to r'}_{\mathbf{\Sigma} A}\,(\alpha \mapsto t)\,(b)$, and the composition structure for the output is the one induced by it (every non-dependent type has a composition structure if it has a homogeneous composition structure).

On $r = r'$, $\mathtt{wcoe}^{z:r \to r'}_{\mathbf{\Sigma} A}\,(-)$ is an η-expanded "inductive identity function". Thus, the path needed to complete the definition of a weak coercion structure is also an η -expanded identity (this is another new elimination rule for suspensions):

$$\frac{\Psi \vdash r, r' : \mathbb{I} \qquad \Psi, z : \mathbb{I}; \phi; \Gamma \vdash A : \texttt{Type} \qquad \Psi; \phi; \Gamma \vdash b : \mathbf{\Sigma}(A\langle r/z\rangle)}{\Psi; \phi; \Gamma \vdash \eta_{\mathbf{\Sigma}A}(b) : \texttt{Path}_{\mathbf{\Sigma}A}\,(\texttt{wcoe}^{z:r\to r'}_{\mathbf{\Sigma}A}\,(b), b)}$$

$$\begin{aligned}
\eta(\texttt{north}) &\equiv \Lambda_.\texttt{north} \\
\eta(\texttt{south}) &\equiv \Lambda_.\texttt{south} \\
\eta(\texttt{merid}_{r_0}(u)) &\equiv \Lambda_.(\texttt{merid}_{r_0}(u)) \\
\eta(\texttt{vhcom}^{r\to r'}_{\mathbf{\Sigma}A}\,(\alpha \mapsto t)\,(b)) &\equiv \Lambda x.\texttt{vhcom}^{r\to r'}_{\mathbf{\Sigma}A}\,(\alpha \mapsto z.\eta(t)\,x, 0/x \mapsto _.\texttt{wcoe}^{z:r\to r'}_{\mathbf{\Sigma}A}\,(b), 1/x \mapsto _.b)\,(\eta(b)\,x)
\end{aligned}$$

The final equation says that η strictly preserves hcoms, using, in the codomain, the fact that $\texttt{Path}_A\,(x, y)$ is fibrant over (x, y) (but not A) given only a homogeneous composition structure in A (to be fibrant over the type A, as above, requires a full composition structure for A).

In our formalized model, we have defined weak coercion for the general pushout of a cospan of functions $f : C \to A$, $g : C \to B$. The definition follows the same steps as above, but the construction of the weak coercion function and η path are a bit more complex, because in the case of the pushout path constructor, it is necessary to adjust by some naturality paths between $f\langle r'/z\rangle(\texttt{coe}^{z:r\to r'}_C(c))$ and $\texttt{coe}^{z:r\to r'}_A(f\langle r/z\rangle(c))$ when $x : \mathbb{I} \vdash f : C \to A$ and $c : C\langle r/z\rangle$.

2.16 Identity Types

Cohen, Coquand, Huber, and Mörtberg [2016] use an idea of Andrew Swan to construct identity types (with a judgemental equality for J on `refl`) from path types. The same definition of the identity type applies in our model, though the definition of J is a bit more complex. Implementing the J elimination rule uses contractibility of singleton types, which in Cohen, Coquand, Huber, and Mörtberg [2016] is proved using a connection. Here, we instead use the Kan operation, and we need to check that we can do so in such a way that singleton contractibility is `refl` on `refl`.

For this section only, we add the following cofibration rules:

$$\frac{\phi, \alpha \vdash_\Psi \beta \quad \phi, \beta \vdash_\Psi \alpha}{\Psi; \phi \vdash \alpha \equiv \beta\ \mathsf{cofib}} \quad \frac{\Psi \vdash \alpha\ \mathsf{cofib} \quad \Psi \vdash \beta\ \mathsf{cofib}}{\Psi \vdash \alpha \wedge \beta\ \mathsf{cofib}} \quad \frac{\phi \vdash_\Psi \alpha \quad \phi \vdash_\Psi \beta}{\phi \vdash_\Psi \alpha \wedge \beta} \quad \frac{\phi \vdash_\Psi \alpha \wedge \beta \quad \Psi; \phi, \alpha, \beta \vdash J}{\Psi; \phi \vdash J}$$

The first is "propositional univalence"—interprovable cofibrations are equal. The next rules say that cofibrations are closed under pullback.

Swan defines the identity type as consisting of paths equipped with cofibrations remembering where the paths are constant:

$$\frac{\Psi;\phi;\Gamma\vdash A\ \mathtt{Type}\quad \Psi;\phi;\Gamma\vdash a_0 : A\quad \Psi;\phi;\Gamma\vdash a_1 : A}{\Psi;\phi;\Gamma\vdash \mathtt{Id}_A(a_0,a_1)\ \mathtt{Type}}$$

$$\frac{\Psi;\phi;\Gamma\vdash p : \mathtt{Path}_A\,(a_0,a_1)\quad \Psi;\phi,\alpha;\Gamma\vdash p\equiv \Lambda\_.a_0 : \mathtt{Path}_A\,(a_0,a_1)}{\Psi;\phi;\Gamma\vdash (\alpha,p) : \mathtt{Id}_A(a_0,a_1)}$$

$$\frac{\begin{array}{l}\Psi;\phi;\Gamma\vdash A\ \mathtt{Type}\\ \Psi;\phi;\Gamma\vdash a_0,a_1 : A\\ \Psi;\phi;\Gamma\vdash p : \mathtt{Id}_A(a_0,a_1)\\ \Psi;\phi;\Gamma,u : \Sigma x{:}A.\mathtt{Id}_A(x,a_1)\vdash C\ \mathtt{Type}\\ \Psi;\phi;\Gamma\vdash c : C[(a_1,\mathtt{refl}_{a_1})/u]\end{array}}{\Psi;\phi;\Gamma\vdash \mathtt{J}_{u.C}(p,c) : C[(a_0,p)/u]}$$

$$\mathtt{J}_{u.C}(p,c)\equiv \mathtt{transport}_{u.C}(\mathtt{scontr}(a_0,p),c)$$

$$\begin{aligned}&\mathtt{com}^{z:r\to r'}_{\mathtt{Id}_A(a_0,a_1)}\,(\alpha\mapsto z.(\alpha_t,t))\,((\alpha_b,b))\\ \equiv\ &((\alpha\wedge(\alpha_t\langle r'/z\rangle))\vee(r=r'\wedge\alpha_b),\mathtt{com}^{z:r\to r'}_{\mathtt{Path}_A\,(a_0,a_1)}\,(\beta\mapsto z.t)\,(b))\end{aligned}$$

We write $\mathtt{refl}_a$ for $(\top,\Lambda\_.a)$, where $\top$ is $0=0$.

In the equation defining composition, the idea is that the composite is constant when the path composite is $t\langle r'/z\rangle$ (on α) and $t\langle r'/z\rangle$ is constant (α_t at r'), or when the path composite is b ($r=r'$) and b is constant (α_b). To boundary-check this, we use the corresponding equations for the path composite, as well as

$$\begin{aligned}\alpha&\vdash((\alpha\wedge(\alpha_t\langle r'/z\rangle))\vee(r=r'\wedge\alpha_b))\equiv(\alpha_t\langle r'/z\rangle)\\ r=r'&\vdash((\alpha\wedge(\alpha_t\langle r'/z\rangle))\vee(r=r'\wedge\alpha_b))\equiv\alpha_b\end{aligned}$$

These follow from propositional univalence for cofibrations, using the fact that $\alpha\vdash\alpha_t\langle r/z\rangle\equiv\alpha_b$ because the original composition problem was well-formed.

The computation rule for $\mathtt{J}$ reduces it to transport and singleton contractibility, which are abbreviations defined as follows. For transport, we define

$$\frac{x : A\vdash C\ \mathtt{Type}\quad (\alpha,p) : \mathtt{Id}_A(a_0,a_1)\quad b : C[a_0/x]}{\mathtt{transport}_{x.C}((\alpha,p),b) := \mathtt{com}^{z:0\to 1}_{C[p\,z/x]}\,(\alpha\mapsto\_.b)\,(b) : C[a_1/x]}$$

The tube is well-typed because $p\,z$ is equal to a_0 on α, because p is constant on α. Observe that this definition is constantly b on α, and α is $\top$ for $\mathtt{refl}$, so transport cancels on $\mathtt{refl}$. Thus, to get the computation rule for $\mathtt{J}$, it suffices to define singleton contractibility in such a way that it sends $\mathtt{refl}$ to $\mathtt{refl}$. For singleton contractibility, write $S(a_1)$ for $\Sigma x{:}A.\mathtt{Id}_A(x,a_1)$. We define

$$\frac{a_0 : A\quad p : \mathtt{Id}_A(a_0,a_1)}{\mathtt{scontr}(a_0,p) : \mathtt{Id}_{S(a_1)}((a_1,\mathtt{refl}_{a_1}),(a_0,p))}$$

by

$$\begin{aligned}&\mathtt{scontr}(a_0,(\alpha,p)) := (\alpha,\Lambda x.(s\langle 0/y\rangle,((x=0\vee\alpha),\Lambda y.s)))\\ &x:\mathbb{I},y:\mathbb{I}\vdash s := \mathtt{hcom}^{1\to y}_A\,(x=0\mapsto\_.a_1,\ x=1\mapsto y.p\,y,\ \alpha\mapsto\_.a_1)\,(a_1)\end{aligned}$$

Since this definition outputs the same cofibration α as the input path, it sends $\mathtt{refl}$ to $\mathtt{refl}$. Thus, $\mathtt{J}$ on $\mathtt{refl}$ cancels.

3 Semantics

The above syntactic presentation is intended to be sound for a model in cubical sets on the Cartesian cube category $\mathbb{C}$, and other cube categories with suitable structure. We have not yet written out a formal translation of the above syntax into the model, but we have constructed the semantic entities that the syntax will be interpreted as. As much of the model as possible is formalized using the Agda proof assistant as the internal logic of cubical sets, following Orton and Pitts [2016].

3.1 Overview of the Formalization

3.1.1 Assumptions

First, we make Agda into a pseudo-extensional type theory by postulating function extensionality and a -1-truncated/squashed disjunction, which we write as $\alpha 1 \vee \alpha 2$. We write Prop for the type of strict propositions, types that satisfy (x y : P) → x = y (the formalization uses universe polymorphism, so really there are propositions in each universe).

The following postulates are a subset of those used in Orton and Pitts [2016], except for Cofib-=, which allows diagonal cofibrations. We postulate an interval type

```
𝕀 : Set
‘0 : 𝕀
‘1 : 𝕀
```

which is non-trivial

```
iabort : ‘0 = ‘1 → ⊥
```

and connected

```
iconnected : (P : 𝕀 → Prop)
  → ((i : 𝕀) → (fst (P i) ∨ (fst (P i) → ⊥)))
  → (((i : 𝕀) → fst (P i)) ∨ ((i : 𝕀) → (fst (P i)) → ⊥))
```

Next, we postulate a notion of cofibration closed under $\forall$, $\vee$, and == in $\mathbb{I}$:

```
isCofib : Set → Set
isCofib∨ : ∀ {α1 α2} → isCofib α1 → isCofib α2 → isCofib (α1 ∨ α2)
isCofib= : ∀ {r r' : 𝕀} → isCofib (r = r')
isCofib∀ : ∀ {α : 𝕀 → Set} → ((x : 𝕀) → isCofib (α x)) → isCofib ((x : 𝕀) → α x)
```

When α is a cofibration and t : α → A is a partial element of A, we write

```
A [ α ↦ t ] = Σ[b : A] (pα : α) → t pα = b
```

for an element of A that restricts on α to t, and we use an analogous binary version of this notation.

Finally, we have the strictification axiom:

```
strictify : {α : Set} {cα : Cofib α} (A : α → Set l) (B : Set l)
  → (i : (pα : α) → Iso B (A pα))
  → Σ[B' : Set l [ α ↦ A ] ]
     Iso B (fst B') [ α ↦ (λ pα → eqIso (snd B' pα) ∘iso i pα) ]
```

This says that given a type B and a partial equivalence with A, we can make a type B' that is isomorphic to B and strictly A on α (and the isomorphism with B restricts to the provided t). This is used to construct Glue types.

3.1.2 Representation of the Kan Operation

Our representation of the Kan operation is analogous to [Orton and Pitts, 2016], adapted to the Kan operation that we use here. First, we define what a composition structure for a family dependent on $\mathbb{I}$ is, with a definition that looks much like the syntactic rule:

```
hasCom : ∀ {l} → (𝕀 → Set l) → Set (lsuc lzero ⊔ l)
hasCom A = (r r' : 𝕀) (α : Set) {_ : Cofib α}
  → (t : (z : 𝕀) → α → A z)
  → (b : A r [ α ↦ t r ])
  → A r' [ α ↦ t r' , (r = r') ↦ ⇒ (fst b) ]
```

We write ⇒ for transport/subst along the Agda equality type (this would be silent in an actual extensional type theory). In this case (fst b) : A r and the partial element needs to have type (r = r') → A r'. Set should be thought of as the universe of non-Kan cubical sets, and this defines what a Kan operation on a cubical set is. Then, for a type A dependent on Γ, a composition structure relative to Γ is a composition structure in the above sense for all paths in Γ:

```
relCom : ∀ {l1 l2} {Γ : Set l1} (A : Γ → Set l2) → Set (lsuc lzero ⊔ l1 ⊔ l2)
relCom {Γ} A = (p : 𝕀 → Γ) → hasCom (A o p)
```

This version matches the filling diagrams we drew in the introduction because p itself can have free variables other than the bound $\mathbb{I}$.

We also define the variant composition operations that play an important role in our definitions:

```
hasHCom : {l : Level} → Set l → Set (lsuc lzero ⊔ l)
hasHCom A = (r r' : 𝕀) (α : Set) {{_ : Cofib α}}
  → (t : (z : 𝕀) → α → A)
  → (b : A [ α ↦ t r ])
  → A [ α ↦ t r' , (r = r') ↦ k (fst b) ]
hasCoe : {l : Level} → (𝕀 → Set l) → Set l
hasCoe A = (r r' : 𝕀)
  → (b : A r)
  → A r' [ (r = r') ↦ ⇒ b ]
relCoe : ∀ {l1 l2} {Γ : Set l1} (A : Γ → Set l2) → Set (l1 ⊔ l2)
relCoe {Γ} A = (p : 𝕀 → Γ) → hasCoe (A o p)
hasWCoe : {l : Level} → (𝕀 → Set l) → Set l
hasWCoe A = (r r' : 𝕀) →
  Σ[f : A r → A r']
    ((e : r = r') → (b : A r) → Path (A r') (f b) (⇒ b e))
relWCoe : ∀ {l1 l2} {Γ : Set l1} (A : Γ → Set l2) → Set (l1 ⊔ l2)
relWCoe {Γ} A = (p : 𝕀 → Γ) → hasWCoe (A o p)
```

An example lemma is

```
coe-from-wcoe : ∀ {l1 l2} {Γ : Set l1} (A : Γ → Set l2)
  → ((x : Γ) → hasHCom (A x))
  → relWCoe A
  → relCoe A
coe-from-wcoe A hcomA wcoeA p s s' b =
  fst (hcomA (p s') ‘0 ‘1 (s = s')
```

```
(λ x → λ s=s' → fst ((snd (wcoeA p s s') s=s') b) x)
((fst (wcoeA p s s') b) ,
  ...) , ...
```

This defines a strict coercion structure from a weak one and hcom, as discussed above. The elided parts are Agda proof terms verifying that the boundary equations necessary for the hcom to be well-formed are true (e.g. at x = 0, fst ((snd (wcoeA p s s') s=s') b) x is equal to (fst (wcoeA p s s') b)), and that the boundary constraints for a coercion operation are satisfied (e.g. on s=s', the above term is equal to b).

3.1.3 Constructions

In the companion code, we define instances of relCom A for Π, Σ, Path, identity, Glue, natural number, boolean, and pushout types. We have not checked the fibrancy of the universe (which must be done externally using current techniques) or the pushout-elim types $[\alpha_1 \mapsto A, \alpha_2 \mapsto B]$, whose fibrancy is immediate when types are represented as maps into a universe of fibrant types.

For example, the overall construction of the Kan operation for Glue, formalizing Section 2.11, has the following Agda types:

```
comGlue : {l1 l : Level} {Γ : Set l1}
  (α : Γ → Cofibs)
  (T : (x : Γ) → fst (α x) → Set l)
  (B : Γ → Set l)
  (f : (x : Γ) (u : fst (α x)) → T x u → B x)
  (comT : relCom (λ (p : Σ[x : Γ] fst (α x)) → T (fst p) (snd p)))
  (comB : relCom B)
  (wCoeGlue : relWCoe (GlueF α T B f))
  → relCom (GlueF α T B f)
comGlue-∀α : ... same assumptions ...
  (p : I → Γ) (p∀α : ∀i (fst o α o p)) →
  ∀ s s' α' {{cα'}} t b →
    fst (comGlue α T B f comT comB wCoeGlue p s s' α' {{cα'}} t b)
  = (coe (! (Glue-α _ {{(snd (α (p s')))}} _ _ _ (p∀α s'))) (fst ((fill-Glue-fiber p p∀α s s' α' {{cα'}} t b))))
GlueF-stable : ... same assumptions
  → comPre θ (comGlue α T B f comT comB wCoeGlue) =
    comGlue (α o θ) (λ z → T (θ z)) (B o θ) (λ z → f (θ z))
      (comPre (λ p → (θ (fst p) , snd p)) comT)
      (comPre θ comB)
      (λ p → wCoeGlue (θ o p))
```

The first says that given Kan types T and B and a weak coercion structure for the Glue type, the glue type has a composition structure. Moreover, on ∀w.α, this restricts to composition in T (the term fill-Glue-fiber ... is the desired composite in T). Finally, the composition structure is stable under substitutions, which is necessary to interpret the syntactic equation $(\texttt{Glue}\,(\alpha[\theta] \mapsto (T, e))\,(B))[\theta] = (\texttt{Glue}\,(\alpha[\theta] \mapsto (T[\theta], e[\theta]))\,(B[\theta]))$.

For suspension and pushout types, we postulate the definition of the type as a cubical set (the intro, elim, and computation rules), and then show that the type has a weak coercion structure, and therefore a composition structure, following the definition in Section 2.15.

To construct identity types using Swan's technique, we additionally postulate

```
isCofib-prop : ∀ {A : Set} → (p q : Cofib A) → p = q
isCofib∧ : ∀ {α} {α' : α → Set} → isCofib α → ((x : α) → isCofib (α' x)) → isCofib (Σ α')
Cofib-propositional-univalence : ∀ {α α'} {{cα : Cofib α}} {{cα' : Cofib α'}}
  → (α → α') → (α' → α) → α = α'
```

3.2 Validating the Axioms

The next step is to argue that the Agda formalization is meaningful in cubical sets. Let C be a finite product category with an object $\mathbb{I}$, with maps $0, 1 : 1 \to \mathbb{I}$ with $0 \neq 1$. In particular C might be $\mathbb{C}$, the Cartesian cube category, which certainly has finite products and such an interval.

First, the logical framework is interpreted in essentially the same way as in Orton and Pitts [2016]. Agda's Π, and Σ are interpreted as the dependent types in any presheaf model. For propositions, we use Agda's Π (for $\forall$) and Σ (for $\wedge$), along with equality types (with postulated function extensionality and uniqueness of identity proofs) and a postulated "squashed" disjunction. Like any presheaf category, cubical sets are a topos, and these propositions can be interpreted using the subobject classifier of the topos (though our Agda development is predicative, so we do not need the full power of this). For natural numbers and booleans, we use Agda datatypes, which can be interpreted as the discrete cubical sets whose 0-cells are these types.

What remains is to validate the axioms specific to cubical sets. Relative to Orton and Pitts [2016], we have changed one axiom (`ax5`, by making $r = r'$ a cofibration), so we need to check that the generalized axiom still works. We have also changed the base category of the presheaves, though most of the theorems in Orton and Pitts [2016] are phrased in sufficient generality that they still apply.

- *Connectedness* says that decidable propositions are constant on the interval:

  $$\forall P : \mathbb{I} \to \texttt{Prop}, (\forall i : \mathbb{I}. P(i) \vee \neg(P(i))) \to (\forall i : \mathbb{I}. P(i)) \vee (\forall i : \mathbb{I}. \neg(P(i)))$$

  In our current formalization, this is used only for defining the Kan operations for strict base types (natural numbers, booleans), but it may also be used for (or implied by other axioms used for) the universe. To see that it is true in $\hat{C}$, the argument in Orton and Pitts [2016] applies to presheaves on any category C that is inhabited (in this case by $\cdot$) and has finite products.

- *Non-triviality:* $0 \neq 1 : \mathbb{I}$. The interval is interpreted as the representable presheaf on the interval $\mathbb{I}$ in C, and $0/x \neq 1/x$ as maps $\Psi \to \mathbb{I}$ in C.

- *Strictification:* (Orton-Pitts $\texttt{ax}_8$) This axiom is used only once, to construct glue types from Σ-types, in such a way that on α they are equal to T. Theorem 6.3 of Orton and Pitts [2016] shows that strictification holds in any presheaf topos, if the cofibrations are decidable (factor through the inclusion of decidable propositions into the subobject classifier). So the result carries over, as long as we show that our cofibrations are decidable.

- Cofibrations: In the Agda formalization, we postulate that cofibrations are closed under $=_{\mathbb{I}}$, $\vee$, and $\forall x : \mathbb{I}.-$. Semantically, the only constraint is that cofibrations are decidable, i.e. that $\texttt{Cof} \rightarrowtail \Omega$ factors through the subobject of decidable propositions $\texttt{Cof} \hookrightarrow \Omega_{dec} \hookrightarrow \Omega$. Thus, as in [Orton and Pitts, 2016], we have many choices for how to interpret cofibrations.

At one extreme, we can take `Cof` to be a "face lattice" closed under only $=_{\mathbb{I}}$ and $\vee$, and observe that $\forall x : \mathbb{I} -$ is admissible for this fragment by a quantifier elimination argument, as in Cohen, Coquand, Huber, and Mörtberg [2016]: define $\forall x.x = x$ to be true, $\forall x.x = r$ to be false if $x \neq r$, $\forall x.r = r'$ to be true if $x \neq r, r'$, and $\forall x.(\alpha \vee \beta) = (\forall x.\alpha) \vee (\forall x.\beta)$. Then $r = r'$ is decidable, because $\hom_{\mathbb{C}}(\Psi, \mathbb{I})$ has decidable equality—equality of maps is just syntactic identity in the $\Psi \vdash r : \mathbb{I}$ judgement. And the $\vee$ of decidable propositions is always decidable. While their paper [Orton and Pitts, 2016] does not consider $\forall w.\alpha$, Orton and Pitts have given an argument that (for the Cohen, Coquand, Huber, and Mörtberg [2016] notion of cofibration) this definition satisfies $\texttt{Cof} \hookrightarrow \Omega_{dec}$—this is essentially a soundness and completeness argument.

At the other extreme, we can take `Cof` to be Ω_{dec} itself, which is closed under $\forall$ for the following reason. Ω_{dec} classifies monomorphisms $m : A \Rightarrow_{\hat{\mathbb{C}}} B$ such that for all Ψ, $A(\Psi) \Rightarrow B(\Psi)$ has a decidable image, i.e. one can decide whether a $b \in B(\Psi)$ is in the image of $m(\Psi)$. Sattler [2017] has shown that cofibrations are closed under $\forall$ iff they are closed under exponentiation by $\mathbb{I}$, i.e. $A \hookrightarrow B$ is a cofibration implies $A^{\hom_{\mathbb{C}}(-,\mathbb{I})} \hookrightarrow B^{\hom_{\mathbb{C}}(-,\mathbb{I})}$ is a cofibration. Here, the class of cofibrations is monos with decidable image, so it suffices to show that these are closed under exponentiation by $\mathbb{I}$. But if $m : A \hookrightarrow B$ has decidable image for all Ψ, then so does $m^{\hom_{\mathbb{C}}(-,\mathbb{I})} : A^{\hom_{\mathbb{C}}(-,\mathbb{I})} \hookrightarrow B^{\hom_{\mathbb{C}}(-,\mathbb{I})}$, because

$$m^{\hom_{\mathbb{C}}(-,\mathbb{I})}(\Psi) : A^{\hom_{\mathbb{C}}(-,\mathbb{I})}(\Psi) \hookrightarrow B^{\hom_{\mathbb{C}}(-,\mathbb{I})}(\Psi) = m(\Psi \times \mathbb{I}) : A(\Psi \times \mathbb{I}) \hookrightarrow B(\Psi \times \mathbb{I})$$

is just a dimension shift. (This argument works for $\forall x : \hom_{\mathbb{C}}(-, J).-$ for any object J of the cube category, as long as the cube category has finite products.)

Relative to Orton and Pitts [2016], we drop `ax3`/`ax4`, which specify connections, and `ax7`, which says that cofibrations are closed under $\wedge$.

3.3 Universes

Using the techniques reported in [Orton and Pitts, 2016], it does not seem possible to formalize a universe in the internal language, so we need to work externally. Here, we repeat some of the construction of fibrant universes from Cohen, Coquand, Huber, and Mörtberg [2016], and argue that it still applies.

Semantic Judgements Assume a universe hierarchy in sets, which we write as Set_i. For a cubical set in universe i, $\Gamma \in \hat{\mathbb{C}}_i$, we define a semantic type in universe i, $Ty_i(\Gamma)$ to be

- A function $A^0 : \forall \Psi \in Ob(\mathbb{C}), \Gamma_\Psi \to Set_i$
- and a function $A^1 : \forall \rho : \Psi_1 \to_{\mathbb{C}} \Psi_2, (\theta : \Gamma_{\Psi_2}) \to A^0_{\Psi_2}(\theta) \to A^0_{\Psi_1}(\theta\langle\rho\rangle)$
- such that A^1 preserves identity and composition.

Given a natural transformation $\theta : \Delta \Rightarrow \Gamma$, and a type $A : Ty_i(\Gamma)$, a type $A[\theta] : Ty_i(\Delta)$ is defined by precomposition.

Cofibrations form a presheaf $Cof \in \hat{\mathbb{C}}$, which at Ψ is the set of cofibrations on Ψ, and with maps acting by substitution/pullback.

Semantic Kan Operation A composition structure on a semantic type $A : Ty_i(\Gamma)$ is

- a function

$$\begin{array}{ll} com : \forall & \Psi : \mathbb{C}, \alpha : Cof_\Psi, r : \Psi \to_\mathbb{C} \mathbb{I}, r' : \Psi \to_\mathbb{C} \mathbb{I}, \\ & \theta : \Gamma(\Psi \times \mathbb{I}), t : A^0_{\Psi.\alpha \times \mathbb{I}}(\Gamma^1(\alpha \times id : \Psi.\alpha \times \mathbb{I} \hookrightarrow \Psi \times \mathbb{I})(\theta)), b : A^0_\Psi(\Gamma^1(r/x)(\theta)) \\ & \to (A^1(\alpha)(b) = A^1(r/x)(t) : A^0_{\Psi.\alpha}(\Gamma^1(\alpha, r/x)(\theta)) \\ & \to A^0_\Psi(\Gamma(r'/x)\theta) \end{array}$$

- which is *uniform*:

$$\begin{gathered} \forall(\rho : \Psi' \to_\mathbb{C} \Psi), A_1(\rho)(com(\Psi, \alpha, r, r', \theta, t, b)) = \\ com(\Psi', Cof^1(\rho)(\alpha), r\langle\rho\rangle, r'\langle\rho\rangle, \Gamma_1(\rho)(\theta), A_1(\rho)(t), A_1(\rho)(b)) \end{gathered}$$

- and such that

$$\begin{array}{l} com(\Psi, \alpha, r, r', \theta, t, b) = t \text{ if } \top \to_{\mathbf{Sub}(\Psi)} \alpha \\ com(\Psi, \alpha, r, r', \theta, t, b) = b \text{ if } r =_{\hom_\mathbb{C}(\Psi,\mathbb{I})} r' \end{array}$$

The only subtlety of this definition is that the "contexts" of α, r, and r' are different in the syntax than in the semantics: in the syntax, these depend on the ambient context $\Psi; \phi; \Gamma$, whereas in the semantics they are in the external context Ψ_0 (in the diagrams in the introduction, the left-hand side of a filling problem is not dependent on the right-hand size). Because of the meaning of $\Psi.\phi.\Gamma$ (Ψ is the representable, $\Psi.\phi$ is a subpresheaf of it, $\Psi.\phi.\Gamma$ is the category of elements of that), any element $(\theta : (\Psi.\phi.\Gamma)_{\Psi_0})$ includes as its first components a map $\rho_\theta : \hom_\mathbb{C}(\Psi_0, \Psi)$ such that $\top \to_{\mathbf{Sub}(\Psi_0)} \phi\langle\rho\rangle$, i.e. a substitution that makes ϕ true. Thus, the interpretation of the syntactic composition is the semantic term

$$[\![\mathtt{com}_A^{z:r \to r'}\,(\alpha \mapsto t)\,(b)]\!] := \Psi_0, (\theta : (\Psi.\phi.\Gamma)_{\Psi_0}) \mapsto com_A(\Psi_0, [\![\alpha]\!]\rho_\theta, [\![r]\!]\rho_\theta, [\![r']\!]\rho_\theta, [\![t]\!]\theta, [\![b]\!]\theta)$$

Because of this, the equations on the semantic composition operation correspond to the syntactic ones: $\top \vdash \alpha$ and $r = r'$ are checked after a substitution ρ that makes ϕ true.

Universes The Hofmann-Streicher presheaf lifting of a universe [Hofmann and Streicher, 1997] is a cubical set $U_i \in \hat{\mathbb{C}}_{i+1}$ defined by

- $U_i(\Psi) = Ty_i(\hom_\mathbb{C}(-, \Psi))$
- $U_i(\rho : \Psi_1 \to_\mathbb{C} \Psi_2)(T : Ty_i(\hom_\mathbb{C}(-, \Psi_2)))) = T[(\rho' \mapsto \rho[\rho'])]$, where $(\rho' \mapsto \rho[\rho'])$ is the natural transformation $\hom_\mathbb{C}(-, \Psi_1) \Rightarrow \hom_\mathbb{C}(-, \Psi_2)$ given by post-composition with ρ.

This universe classifies semantic types, in the sense that

$$Tm(\Gamma; U_i) \cong Ty_i(\Gamma)$$

For example, given $A : Ty_i(\Gamma)$, we can define a natural transformation (semantic terms of a non-dependent type are natural transformations) $(code(A)) : \Gamma \Rightarrow U_i$ whose components are

$$\begin{array}{l} (code(A))_\Psi(\theta : \Gamma_\Psi)\,(\rho : \Psi_2 \to \Psi) = A^0_{\Psi_2}(\theta\langle\rho\rangle) \\ (code(A))_\Psi(\theta : \Gamma_\Psi)\,(\rho_{12} : \Psi_1 \to \Psi_2)(a : A^0_{\Psi_2}(\theta\langle\rho\rangle)) = A^1(\rho_{12})(a) : A^0_{\Psi_1}(\theta\langle\rho\rangle\langle\rho_{12}\rangle) \end{array}$$

I.e. the code of A at $\theta : \Gamma_\Psi$ is made "polymorphic" by delaying the cube-category substitution into θ. Conversely, $El : Ty_i(U_i)$ is defined by

$$\begin{array}{l} El^0_\Psi(A : Ty(\hom_{\mathbb{C}}(-, \Psi))) = A^0_\Psi(id_\Psi) \\ El^1(\rho : \Psi \to \Psi')(A : Ty(\hom_{\mathbb{C}}(-, \Psi')))(a : A^0_{\Psi'}(id'_\Psi)) = A_1(\rho)(id_\Psi)(a) : A^0_\Psi(\rho) \end{array}$$

That is, converting a code to a type instantiates the "polymorphic" function with the identity.

We write $FTy_i(\Gamma)$ for the set of pairs $(A : Ty_i(\Gamma), com_A)$ where com_A is a composition structure for A. Natural transformations $\theta : \Delta \Rightarrow \Gamma$ act on this, sending a type A to $A[\theta]$, with $com_{A[\theta]}$ also given by precomposition into com_A.

Then the universe F_i of fibrant types is defined by

$$F_i(\Psi) := FTy_i(\hom_{\mathbb{C}}(-, \Psi))$$

with reindexing along $\Psi_1 \to_{\mathbb{C}} \Psi_2$ again given by substitution along the natural transformation $\hom_{\mathbb{C}}(-, \Psi_1) \Rightarrow \hom_{\mathbb{C}}(-, \Psi_2)$. This classifies fibrant types:

$$Tm(\Gamma; F_i) \cong FTy_i(\Gamma)$$

Interpreting the internal statements proved in Agda entails that F_i is closed under Π, Σ, path, boolean, natural number, and glue types. This is because an internal language closed term $\cdot \vdash A : \Gamma \to \mathsf{Set}$ is interpreted as $Tm(\Gamma; U_i)$, and the interpretation of an internal composition structure (with some massaging by the above isomorphisms) gives the second component of the element of F_i, the semantic composition structure. The internal construction of a composition structure is automatically uniform, because reindexing externally corresponds to internal substitution, and all internal terms commute with substitution.

Because the universe of fibrant types is closed under glue types, we can define a composition structure on F_i itself by the semantic analogue of the glue type indicated in Section 2.12, so there is a code for F_i in F_{i+1}. Thus, we have fibrant universes of fibrant types.

We expect the univalence axiom to be an internally provable theorem in the syntactic type theory (above, we showed the equivalence-to-path function); this, plus a soundness theorem for the syntax, will show that these universes are univalent.

3.4 Connection to the Connections Kan Operation

The assumptions of the argument in Section 3.2 are satisfied by the connections cube category, so we can also interpret our construction there, and in the de Morgan cube category Cohen, Coquand, Huber, and Mörtberg [2016] (in both cases, the cube category has finite products and an interval with appropriate structure). Consider the de Morgan case, and take $r =_{\mathbb{I}} r'$, pushouts, pullbacks, and $\forall x : \mathbb{I}.-$ to be cofibrations for both Kan operations. This is a setting that includes the structure of both Cohen, Coquand, Huber, and Mörtberg [2016] and our model.

The Cohen, Coquand, Huber, and Mörtberg [2016] Kan operation is $\mathtt{com}_A^{z:0\to 1}\,(\alpha \mapsto t)\,(b)$. We call a type equipped with such an operation endpoint-Kan. We call a type that is Kan in the sense of Definition 1 diagonal-Kan.

If a type is diagonal-Kan, then it is immediately endpoint-Kan as a special case. Conversely, using connections and the reversal, we define

$$\begin{aligned}
&\mathit{if}(r = 0, r_1, r_2) := (r \vee r_1) \wedge ((1 - r) \vee r_2) \wedge (r_1 \vee r_2) \\
&\mathit{if}(0 = 0, r_1, r_2) \equiv r_1 \\
&\mathit{if}(1 = 0, r_1, r_2) \equiv r_2 \\
&\mathit{if}(r = 0, r_1, r_1) \equiv r_1
\end{aligned}$$

Then diagonal Kan composition $\mathtt{com}_A^{z:r \to r'}\,(\alpha \mapsto t)\,(b)$ is derived as

$$\mathtt{com}^{z:0\to 1}_{A\langle \mathit{if}(z=0,r,r')/z\rangle}\,(\alpha \mapsto z.t\langle \mathit{if}(z = 0, r, r')/z\rangle, (r = r') \mapsto \_.b)\,(b)$$

Notice the use of a diagonal cofibration—without it, this definition would not satisfy the strict $r = r'$ constraint without regularity.

Because any two composites of the same filling problem are path-equal, these two translations must be mutually inverse (up to paths), and this should lead to an equivalence between the universe of endpoint-Kan types and the universe of diagonal-Kan types, in a setting where both exist.

References

T. Altenkirch and A. Kaposi. A syntax for cubical type theory. Available from `http://www.cs.nott.ac.uk/~txa/publ/ctt.pdf`, 2014.

T. Altenkirch, C. McBride, and W. Swierstra. Observational equality, now! In *Programming Languages meets Program Verification Workshop*, 2007.

C. Angiuli and R. Harper. Computational higher type theory II: dependent cubical realizability. arXiv:1606.09638, 2017.

C. Angiuli, R. Harper, and T. Wilson. Computational higher type theory I: abstract cubical realizability. arXiv:1604.08873, 2016.

C. Angiuli, R. Harper, and T. Wilson. Computational higher-dimensional type theory. In *ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, 2017a.

C. Angiuli, K.-B. Hou (Favonia), and R. Harper. Computational higher type theory III: univalent universes and exact equality. arXiv:1712.01800, 2017b.

S. Awodey. A cubical model of homotopy type theory. arXiv:1607.06413, 2016a.

S. Awodey. Notes on cubical sets. Available from `https://github.com/awodey/math/blob/master/Cubical/cubical.pdf`, 2016b.

J.-P. Bernardy, T. Coquand, and G. Moulin. A presheaf model of parametric type theory. In *Mathematical Foundations of Programming Semantics*, 2015.

M. Bezem, T. Coquand, and S. Huber. A model of type theory in cubical sets. Preprint, September 2013.

G. Brunerie and D. R. Licata. A cubical infinite-dimensional type theory. Talk at Oxford Workshop on Homotopy Type Theory, November 2014.

U. Buchholtz and E. Morehouse. Varieties of cubical sets. In *Relational and Algebraic Methods in Computer Science: 16th International Conference, RAMiCS 2017, Lyon, France, May 15-18, 2017, Proceedings*. 2017.

D.-C. Cisinski. Les préfaisceaux comme modèles des types d'homotopie. *Astérisque, Soc. Math. France*, Volume 308, 2006.

C. Cohen, T. Coquand, S. Huber, and A. Mörtberg. Cubical type theory: a constructive interpretation of the univalence axiom. arXiv:1611.02108, To be published in Post-proceedings of the 21st International Conference on Types for Proofs and Programs, TYPES 2015, 2016.

T. Coquand. Variations on cubical sets. Talk at Oxford Workshop on Homotopy Type Theory. Available from `http://www.cse.chalmers.se/~coquand/comp.pdf`, November 2014a.

T. Coquand. Variations on cubical sets (diagonals version). Available from `http://www.cse.chalmers.se/~coquand/diag.pdf`, 2014b.

T. Coquand. Higher inductive types as cubical sets. Available from `http://www.cse.chalmers.se/~coquand/hit1.pdf`, 2015.

N. Gambino and C. Sattler. The frobenius condition, right properness, and uniform fibrations. *Journal of Pure and Applied Algebra*, 221(12):3027 – 3068, 2017.

M. Hofmann. *Extensional Concepts in Intensional Type Theory*. PhD thesis, University of Edinburgh, 1995.

M. Hofmann and T. Streicher. Lifting grothendieck universes. Available from `http://www.mathematik.tu-darmstadt.de/~streicher/NOTES/lift.pdf`, 1997.

S. Huber. Canonicity for cubical type theory. arXiv:1607.04156, 2016.

V. Isaev. Homotopy type theory with an interval type. Available from `https://valis.github.io/doc.pdf`, 2014.

I. Orton and A. M. Pitts. Axioms for modelling cubical type theory in a topos. In *Computer Science Logic*, 2016.

A. M. Pitts. Nominal Presentation of Cubical Sets Models of Type Theory. In *20th International Conference on Types for Proofs and Programs (TYPES 2014)*. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2015.

A. Polonsky. Internalization of extensional equality. Available from `http://arxiv.org/abs/1401.1148`, 2014.

E. Riehl and M. Shulman. A type theory for synthetic ∞-categories. arXiv:1705.07442, 2017.

C. Sattler. The equivalence extension property and model structures. arXiv:1704.06911, 2017.

The Univalent Foundations Program, Institute for Advanced Study. *Homotopy Type Theory: Univalent Foundations Of Mathematics*. Available from `homotopytypetheory.org/book`, 2013.

V. Voevodsky. A very short note on homotopy λ-calculus. *Unpublished*, page 1–7, September 2006. URL `http://www.math.ias.edu/vladimir/files/2006_09_Hlambda.pdf`.