Mathematical Structures in Computer (2019), 1–00
doi:10.1017/xxxxx

CAMBRIDGE
UNIVERSITY PRESS

ARTICLE

Syntax and Models of Cartesian Cubical Type Theory

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

Carnegie Mellon University cangiuli@cs.cmu.edu
Stockholm University guillaume.brunerie@gmail.com
University of Gothenburg Thierry.Coquand@cse.gu.se
Carnegie Mellon University rwh@cs.cmu.edu
University of Minnesota kbh@umn.edu
Wesleyan University dlicata@wesleyan.edu

(Received xx xxx xxx; revised xx xxx xxx; accepted xx xxx xxx)

1. Introduction

Cubical type theories are a family of formal systems for Homotopy Type Theory/Univalent Foundations (Voevodsky, 2006; The Univalent Foundations Program, Institute for Advanced Study, 2013). Unlike type theories that express univalence and higher inductive types as axioms, cubical type theories satisfy the *canonicity* property stating that closed terms compute as programs, providing a constructive justification of the univalence axiom. Moreover, because they provide additional judgemental equalities, and an improved syntax for higher inductive types, cubical type theories allow a more convenient syntax for synthetic homotopy theory.

1.1 Constructive cubical models and type theories

Voevodsky's model of univalence in simplicial sets makes essential use of classical logic (Kapulkin and Lumsdaine, 2021). Because the constructive nature of type theories is one of their major philosophical and practical benefits, a number of researchers in homotopy

type theory were drawn to the problem of finding a constructive model of the univalence axiom.

The BCH model After an early no-go theorem for constructive simplicial set models (Bezem and Coquand, 2015), Bezem et al. (2014, 2018) gave a constructive model of univalent type theory in cubical sets, which are presheaves over the free monoidal category $(C, \otimes, \cdot)$ generated by an interval object $\mathbb{I}$, face maps $0, 1 : \cdot \to \mathbb{I}$, degeneracies $\mathbb{I} \to \cdot$, and a symmetry involution $\mathbb{I} \otimes \mathbb{I} \to \mathbb{I} \otimes \mathbb{I}$. In this and other cubical set models the n-dimensional cells of a type/presheaf A are given by the set $A(\mathbb{I}^{\otimes n})$, and their boundaries are given by restriction along face maps. Types are moreover equipped with *Kan operations* assuring that their n-cells form a higher groupoid, which is necessary to validate the identity elimination rule.

These Kan operations are inspired by the homotopy-theoretic Kan condition—that for any "n-cube missing one face" there is a complete n-cube that "fills" it (i.e., restricts to that configuration) (Kan, 1955)—but are stronger in that (1) they apply to a *broader class* of partial cube configurations, (2) they are given by an *operation* and not only an existence property, and (3) the operation is *uniform*, or natural in the input configuration. The latter condition is crucial for circumventing the constructivity issue experienced in simplicial models.

Bezem et al. implemented an evaluator called `cubical`, which computes terms of univalent type theory in an operationalized version of their model.[a] However, this approach does not produce a better-behaved syntax, because the interpretation of a term in the model is often not expressible in the original theory. For instance, the model interprets identity elimination using Kan operations, which have no syntactic counterpart in ordinary homotopy type theory. A natural next step is therefore to find a *syntactic* presentation of the cubical model that can capture some of the new equations validated by the model as judgemental equalities.

There are a few reasons why this strategy appears plausible. We can make $\mathbb{I}$ explicit in the syntax using variables ranging over its Yoneda embedding as a cubical set; higher cells are then internalized as certain function spaces known as *path types*. Furthermore, the uniform Kan operations are strictly stable under substitutions, making them a good fit for the syntax of type theory. However, there are also a few difficulties. Because the cube category is only monoidal, interval variables are *substructural* (specifically, lacking contraction) as in nominal logic (Pitts, 2015); additionally, without contraction, it is unclear how to express eliminators for higher inductive types.

Geometrically, contraction of interval variables computes a diagonal of an n-cube as an $(n-1)$-cube. Adding contraction/diagonals to the cube category of Bezem et al. yields the *Cartesian cube category*, the free finite product category on an interval object (i.e., the monoidal product $\otimes$ becomes a Cartesian product $\times$). In addition to the syntactic benefits described above, the Cartesian cube category is also natural from a semantic point of view. Indeed, Awodey has encouraged its investigation since 2013 due to its good mathematical properties: it is the classifying topos of bipointed sets; unlike the free symmetric monoidal cube category, geometric realization of Cartesian cubical sets preserves products; and the interval is atomic (i.e., exponentiation by the interval has a right adjoint) (Awodey, 2016). The Cartesian cube category is also a strict test category, meaning, roughly, that presheaves on it (classically) admit a model structure with the same homotopy theory as simplicial sets/topological spaces (though this model structure's fibrations may not coincide with the fibrations used in the cubical type theory or its cubical sets model); Buchholtz and Morehouse (2017) develop a thorough investigation of which cube categories are (strict) test categories.

[a] https://github.com/simhu/cubical

The De Morgan (CCHM) model The constructions of Bezem et al. do not work in the presence of diagonals, because the Kan operations associated to several type formers are not natural in diagonal maps. In 2014 Coquand proposed two alternate uniform Kan operations, differing in what partial cube configurations are allowed: the first was intended for a model in Cartesian cubical sets (Coquand, 2014*b*) and the second for De Morgan cubical sets, which extend the Cartesian cubes with an additional form of degeneracy known as *connections* $\mathbb{I} \times \mathbb{I} \to \mathbb{I}$ (logically corresponding to meets and joins) as well as a *reversal* map $\mathbb{I} \to \mathbb{I}$ (Coquand, 2014*a*). Notably, the latter Kan operation builds in a *regularity* condition which allows path types to strictly model Martin-Löf identity types.

In mid-2014 several syntactic type theories with interval variables were developed: Bernardy et al. (2015) developed a presheaf type theory for polymorphism, Coquand (2014*a*) developed a type-theoretic presentation of the De Morgan model, and Brunerie and Licata (2014) developed a type-theoretic presentation of the incomplete Cartesian model. Isaev (2014) developed an interval-based type theory with diagonals and one connection, which has since evolved into the Arend proof assistant.[b]

Our attempts in 2014–2015 to adapt a proposed construction of univalent universes in the De Morgan model to the Cartesian model revealed that the former did not in fact satisfy the required regularity condition.[c] Following this discovery Cohen et al. (2018) ("CCHM") dropped the regularity condition from the De Morgan model and corrected it to obtain a constructive model of univalent type theory and a De Morgan cubical type theory containing univalent universes closed under Π, Σ, path, natural number, circle, and propositional truncation types for which Huber (2018) proved canonicity. That type theory was implemented in the `cubicaltt` prototype proof assistant[d] and later integrated into Cubical Agda (Vezzosi et al., 2019). Two machine-checked formalizations of this model have been developed. At the suggestion of the fifth author Bickford (2020) gave a direct representation of cubical sets in Nuprl (Constable et al., 1986), verifying that the model can be developed in a constructive metatheory. Orton and Pitts (2016, 2018); Licata et al. (2018); Orton (2019*b*,a) use a more abstract technique, in which Agda is used as the internal language of cubical sets.

The Cartesian model Despite this progress on the De Morgan model, a Cartesian model (and type theory) remained elusive. The constructions of Cohen et al. cannot be adapted to the Cartesian setting, in this case because the Kan operations associated to several type formers make essential use of connections. Awodey (2018*b*) constructed a model of identity types in Cartesian cubical sets as path types using a *regular* uniform Kan composition operation, but this model has not yet been extended to univalent universes.

Angiuli et al. (2016); Angiuli and Harper (2017); Angiuli et al. (2017*a*) made progress by developing a *computational* Cartesian cubical type theory in the sense of Nuprl (Constable et al., 1986), in which types are construed as partial equivalence relations that specify the evaluation behavior of untyped programs. This model validates the rules of a formal type theory, and yields a a strict, rather than homotopy, canonicity result for a Cartesian cubical type theory with Π, Σ, path, boolean, circle, and "isovalence" types, the latter being a variant of univalence for strict isomorphisms (pairs of functions that compose to the identity up to judgemental equality, not up to paths).

In Spring 2017 Angiuli et al. ("AFH") discovered how to construct univalent universes in the Cartesian model by extending the Kan operation to allow open boxes with fixed diagonals (i.e., adding the diagonal $\mathbb{I} \to \mathbb{I} \times \mathbb{I}$ as a cofibration). Using this idea, Angiuli et al. (2017*b*, 2018b); Angiuli (2019) define a computational Cartesian cubical type theory with

[b] `https://arend-lang.github.io/`
[c] See the post `https://goo.gl/btFxZ4` from 5/31/2015 on the Homotopy Type Theory mailing list.
[d] `https://github.com/mortberg/cubicaltt`

a univalent universe hierarchy as well as an extensional equality judgement internalized as an equality pre-type. This type theory was implemented by Angiuli et al. (2018*a*) in the **RedPRL** proof assistant.[e]

In this paper, we define a formal Cartesian cubical type theory that abstracts from the computational semantics, along with a machine-verified Cartesian cubical sets model. This work completes the formalism given by Brunerie and Licata (2014) and the constructive model begun by Coquand (2014*b*). Our work here complements AFH (Angiuli et al., 2017*b*) by showing that the definition of univalent universes in the Cartesian setting admits a broader class of mathematical models, including not only Cartesian cubical sets, but also other presheaf categories, such as ones with additional structure on the interval (e.g. connections, reversals) or additional cofibrations. The definitions of the Kan operations can become quite involved, and while AFH contains extremely detailed on-paper proofs of correctness, the model we present here has been formalized in Agda, lending additional confidence in the correctness of the ideas. Presentationally, we make some choices that expose some similarities with the definitions in the De Morgan model,[f] and disentangle the construction of univalent and Kan universes from an additional feature of AFH, where the Kan composition operation disallows false cofibrations, avoiding the "empty system compositions" that proliferate in formalizations. Since the original development of this work, versions of Cartesian cubical type theory close to the one described in this paper have been implemented in the prototype proof assistants `yacctt`[g] and `redtt`.[h], and the ability to add new interval structure and cofibrations has been used to define a bicubical directed type theory as an extension of this work (Weaver and Licata, 2020).

Notwithstanding the existence of the CCHM model, we believe Cartesian cubical type theory without connections is worth studying for several reasons. First, we found it mathematically interesting to investigate whether connections are necessary to obtain a constructive model of univalence, or whether the technically simpler Cartesian cube category suffices. But aside from curiosity, there are two important reasons to pursue different cubical type theories. One reason is efficiency of implementation, with the goal of (for instance) running Brunerie's calculation of $\pi_4(\mathbb{S}^3) \cong \mathbb{Z}/2\mathbb{Z}$, which none of the existing prototypes are able to do. There are some time-consuming aspects of the implementation of De Morgan model (e.g. checking equality of "systems" in the presence of connections) that are not necessary in the Cartesian model, so it is at least plausible that avoiding connections might be useful for efficiency. A second reason is that cubical type theories differ in their range of models; we would eventually like a cubical type theory that, like axiomatic homotopy type theory, interprets in all ∞-toposes (Shulman, 2019), and it seems plausible that such an interpretation might be easier when less structure on the cube category is demanded by the type theory. A related question is whether the model structure on cubical sets defined from the type theory (Sattler, 2017) is Quillen equivalent to spaces (e.g. simplicial sets). While this is not true (Sattler, 2018) for the Kan composition operation we use here, Awodey, Cavallo, Coquand, Riehl, Sattler have shown that it is true for an "equivariant" extension of our Kan operation (Riehl, 2019). The formal cubical type theory that we define in this paper should interpret in this equivariant model, allowing a translation of theorems proved in it to facts about a standard notion of spaces. (The De Morgan model is also not equivalent to spaces, but the question is still open for the model with connections but no reversal.) Because of these ongoing issues in proof

[e] https://github.com/RedPRL/sml-redprl

[f] We describe Kan composition as a single operation, rather than decomposing it into two simpler operations of coercion and homogeneous composition; we use the "glue type" construction of to simultaneously make the universes univalent and Kan, rather than treating composition in the universe as a primitive.

[g] https://github.com/mortberg/yacctt

[h] https://github.com/RedPRL/redtt

assistant design and semantics, we believe it is worth documenting all of the variants of cubical type theory that support univalence and higher inductive types.

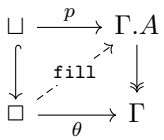
1.2 Background on Kan operations

To explain the main aspects of our contribution in more technical detail, we must delve into the details of cubical type theory and the syntax and semantics of uniform Kan operations. Our understanding of the semantics was particularly influenced by Awodey (2018*b*); Gambino and Sattler (2017); Orton and Pitts (2016); Sattler (2017).

Basic syntax of cubical type theory The judgements of cubical type theory are indexed by a context Ψ of interval (or dimension) variables $x : \mathbb{I}$, whose length expresses the dimension of the judgement. In the case of the typing judgement $\Psi; \Gamma \vdash a : A$, the points of A are given by $\cdot; \Gamma \vdash a : A$, lines by $x : \mathbb{I}; \Gamma \vdash a : A$, squares by $x : \mathbb{I}, y : \mathbb{I}; \Gamma \vdash a : A$, and so on. Semantically, Ψ is an object of a cube category $\mathbb{C}$, here the Cartesian cube category (the free finite product category on an interval object $\mathbb{I}$ with maps $\cdot \vdash 0 : \mathbb{I}$ and $\cdot \vdash 1 : \mathbb{I}$). Closed types are presheaves on $\mathbb{C}$ (objects of $\hat{\mathbb{C}} := \mathbf{Sets}^{\mathbb{C}^{op}}$). In the simple case where Γ is empty and a, but not A, mentions variables from Ψ, the judgement $\Psi \vdash a : A$ represents a natural transformation from $\hom_{\mathbb{C}}(-, \Psi)$ to A, and hence, by the Yoneda lemma, an element of $A(\Psi)$.

Substitution of the special symbols 0 and 1 for interval variables, which we write $a\langle 0/x\rangle$ and $a\langle 1/x\rangle$, corresponds to the action of the presheaf A on the endpoint maps $0, 1 : \cdot \to \mathbb{I}$. Weakening, exchange, and contraction for interval variables correspond to the presheaf action on 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, allowing us to lift their rules to arbitrary interval context Ψ; for example, Π types support λ and application in every $\Psi; \Gamma$. This generalization gives many constructions on paths, such as `ap` and function extensionality, as special cases of the rules for the basic type constructors.

Types as Kan fibrations To express that types behave like spaces (as in homotopy type theory), we require types to be fibrations with respect to the paths given by maps from the dimension variables/interval object $\mathbb{I}$, by adding a Kan filling operation. The classical Kan condition (Kan, 1955) states that any "cube missing one face" (e.g., one endpoint of a line, or three sides of a square) can be "filled" to a complete cube with the given faces on its boundary. This can be rephrased as asserting that commuting squares as indicated in the left diagram admit diagonal lifts:



$$\frac{\sqcup; \Gamma \vdash p : A}{\square; \Gamma \vdash \mathtt{fill}_A(p) : A} \qquad \sqcup; \Gamma \vdash \mathtt{fill}_A(p) \equiv p : A$$

The outer square specifies a fibration/dependent type (on the right), a filling problem (on the left, e.g. the inclusion of a "cube missing one face" into a whole cube), a "whole" shape in Γ (on the bottom), and a "partial" shape in $\Gamma.A$ (on the top) lying over the whole shape (by commutativity). The dashed line indicates that the filler (on the diagonal) exists and lies over the provided whole shape (because the lower triangle commutes) and agrees with the partial shape (because the upper triangle commutes). The inference rule on the right expresses the same principle as a syntactic Kan filling operation—which corresponds semantically to requiring chosen lifts rather than a mere existence property. The equation $\sqcup; \Gamma \vdash \mathtt{fill}_A(p) \equiv p$ expresses the commutativity of the top triangle. The bottom map θ

does not appear in the inference rule, but is implicit in the substitution principles of the type theory; this is because the syntactic filling rule applies not only to the type A, but also to any substitution instance $A[\theta]$. By the usual definition of context extension, any map $\Delta \to \Gamma.A$ whose projection to Γ is some $\theta : \Delta \to \Gamma$ is equivalently a map $\Delta \to \Delta.A[\theta]$ that is a section of the projection to Δ, i.e. a term $\Delta \vdash t : A[\theta]$. Thus, the map p on the top of the diagram can be regarded as a map $\sqcup \to \sqcup.A[\theta]$, and the diagonal filler is a map $\square \to \square.A[\theta]$ (both sections of the projection). Syntactically, the former corresponds to a term $\sqcup; \cdot \vdash p : A[\theta]$, and the latter to $\square; \cdot \vdash \mathtt{fill}_{A[\theta]}(p) : A[\theta]$, which is constructed by the filling operation for $A[\theta]$. This shows that filling problems as given in the diagram can be derived from the inference rule. In addition to being closed under substitutions for Γ, the inference rule is also implicitly closed under substitution for the "shape" context (represented here by $\square$), so its interpretation in cubical sets (as in Cohen et al. (2018, Section 8.2)) requires a bit more than the diagram at the left. Despite this small mismatch, for this introduction, we will pair diagrams and inference rules like these to explain some refinements of this general idea of Kan operations.

The fact that the rule applies to any type expresses that all types are *fibrant* or *Kan* in this sense.

Formulas for cofibrations Next, we discuss how cubical type theories choose and represent the filling problems indicated by $\sqcup \hookrightarrow \square$ above. Semantically, the fibrations considered in cubical type theories correspond (roughly) to those in *cofibrantly generated* model categories. This means that one first chooses a class of *generating cofibrations* and *generating trivial cofibrations*, which describe the shapes of allowed filling problems. The fibrations are then defined to be those types that have lifts (as described above) for any generating trivial cofibration $\sqcup \hookrightarrow \square$ on the left-hand side. Semantically, cofibrations are typically subobjects/monomorphisms (e.g. in classical Cisinski (2006) model structures they are all monomorphisms), while trivial cofibrations are typically contractible objects (one cannot take *all* subobjects of cubes as filling problems—for example filling against the inclusion of the two endpoints into a line would connect any two points by a path). In the classical Kan condition, a generating trivial cofibration is a "cube missing one face" included into the whole cube. This can be viewed as a cofibration part consisting of two faces in every direction *except* one (a "tube"), together with a single face in the remaining ("filling") direction (a "base" or "cap").

Syntactically, the subobjects representing cofibrations and trivial cofibrations can be presented using predicates on the interval variable context Ψ, an approach introduced by Cohen et al. (2018) and developed by Orton and Pitts (2016); Birkedal et al. (2018); Riehl and Shulman (2017). For example, the filling problem (trivial cofibration) given by the inclusion of the left, right, and bottom faces of a square into the square is represented by the formula $(x = 0 \vee x = 1) \vee y = 0$, where in this case the "tube" is the vertical sides $(x = 0 \vee x = 1)$, while the "cap" is the base $y = 0$. The Kan filling operation specialized to this case is

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

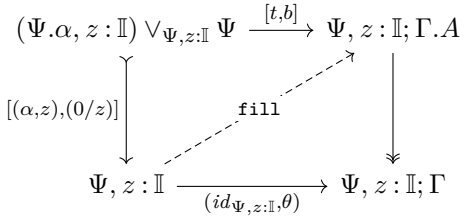
$$x : \mathbb{I}, y : \mathbb{I}; x = 0; \Gamma \vdash \mathtt{fill}_A([[t_0, t_1], b]) \equiv t_0 : A$$

$$x : \mathbb{I}, y : \mathbb{I}; x = 1; \Gamma \vdash \mathtt{fill}_A([[t_0, t_1], b]) \equiv t_1 : A$$

$$x : \mathbb{I}, y : \mathbb{I}; y = 0; \Gamma \vdash \mathtt{fill}_A([[t_0, t_1], b]) \equiv b : A$$

The premises demand a y-path t_0 at $x = 0$ (left), a y-path t_1 at $x = 1$ (right), and an x-path b at $y = 0$ (bottom) that agree on the corners. These three paths (grouped together using $[-,-]$) constitute an element of A in the context $x : \mathbb{I}, y : \mathbb{I}; (x = 0 \vee x = 1) \vee y = 0$, or "the square restricted to be the left, right, or bottom side." The equations then assert that `fill` produces a square whose left, right and bottom are t_0, b, t_1 respectively, as a consequence of a more general equation that when $(x = 0 \vee x = 1) \vee y = 0$ is true the square is equal to $[[t_0, t_1], b]$, along with an equation that when any particular disjunct is true, $[[t_0, t_1], b]$ is equal to the appropriate one of t_0, t_1, b.

In general pairing an arbitrary cofibration α with an endpoint inclusion in a *separate* filling dimension z gives a valid trivial cofibration, because the single cap face in the z direction will connect the tube given by α. The classical Kan case is recovered by taking α to be $x = 0 \vee x = 1$ for every dimension x except z. In this generalized form, the Kan operation allows filling a $\Psi, z : \mathbb{I}$ cube if we are given (1) the tube sides t specified by α, which may depend on the filling direction z, and (2) a cap b at $z = 0$, such that (3) t and b are compatible on both α and $z = 0$. More formally, we have:

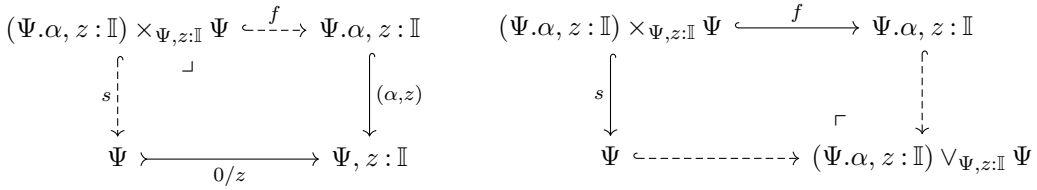


$$\frac{\begin{array}{l}\Psi, z : \mathbb{I}; \Gamma \vdash A\ \mathtt{Type} \\ \Psi.\alpha, z : \mathbb{I}; \Gamma \vdash t : A \\ \Psi; \Gamma \vdash b : A\langle 0/z\rangle \\ \Psi.\alpha; \Gamma \vdash t\langle 0/z\rangle \equiv b : A\langle 0/z\rangle\end{array}}{\Psi, z : \mathbb{I}; \Gamma \vdash \mathtt{fill}_A(t, b) : A}$$

$$\Psi.\alpha, z : \mathbb{I}; \Gamma \vdash \mathtt{fill}_A(t, b) \equiv t : A$$

$$\Psi; \Gamma \vdash (\mathtt{fill}_A(t, b))\langle 0/z\rangle \equiv b : A$$

In the diagram on the left the upper-left object is the pushout of the pullback of (α, z) and $0/z$, which specifies the configuration of a tube and cap that fit together as described informally above. That is, regarding α as a subobject $\Psi.\alpha \hookrightarrow \Psi$, we take the pullback on the left, and then the pushout on the right:



The pushout includes into $\Psi, z : \mathbb{I}$ by the universal property of the pushout applied to (α, z) and $0/z$ (the pushout corner map). The syntactic rule on the right expands the universal property for mapping out of this pushout: we must give a tube t that depends on $z : \mathbb{I}$ but restricted to α, along with a cap b at $z = 0$, which agree on the pullback, i.e. on α and when $z = 0$.[i]

This formulation of the Kan operation makes it clear that the choice of cofibrations is a parameter that can be varied when attempting a cubical model. For example, while the classical Kan operation takes α to be $x = 0 \vee x = 1$ for every dimension x in Ψ, the refinement by Bezem et al. (2014), allowing degeneracies of filling problems as filling problems,

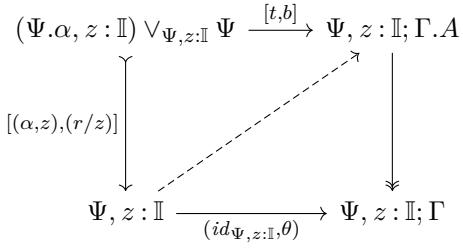
[i] The required semantic filling condition is actually more general than the diagram here, which illustrates a special case that corresponds more directly to the syntactic rule. In general, the left side of the diagram may take place in a different context Ψ'; that is, the left map is $(\Psi'.\alpha, z : \mathbb{I}) \vee_{\Psi', z:\mathbb{I}} \Psi' \to \Psi', z : \mathbb{I}$ and the bottom map may not be of the form (id, θ). In syntax, the general case is obtained by precomposing the given rule with a dimension substitution. We make the same simplification in subsequent diagrams.

corresponds to allowing α to be a disjunction of such pairs for some, but not necessarily all, variables in Ψ. In syntax, this allows the above inference rule to be weakened to a larger interval variable context without changing the raw term. Cohen et al. (2018) additionally allow conjunctions of formulas as cofibrations, which geometrically corresponds to allowing faces more than one dimension lower than the result of the filling problem. While in a classical setting the cofibrations can be taken to be all monomorphisms in the presheaf category, in a constructive setting they must be decidable (in a sense explained below) to allow the definition of "glue" types (Cohen et al., 2018; Orton and Pitts, 2018).

The additional *uniformity* constraint introduced by Bezem et al. (2018) states that the action of any cube map into Ψ commutes with the filling operation. For example, given a filling problem whose face is also a filling problem, the latter's filler is the face of the former's filler. This corresponds to the typical syntactic rule for substitution for the free interval variables in `fill`—i.e. when substituting for an interval variable other than z in $\mathtt{fill}_A(t, b)$, replace all occurrences of that variable in A, t, b with the indicated interval term. For this introductory discussion, we elide this uniformity constraint; see (Awodey, 2018*b*; Gambino and Sattler, 2017) for a semantic analysis.

Generalized trivial cofibrations Thus far, we have identified two points of variations between different cubical models: the choice of cube category (symmetric monoidal, Cartesian, with connections, ... as discussed in Section 1.1), and the choice of cofibrations. A third is the choice of trivial cofibrations. For example, one might also allow $1/z$ in place of $0/z$ in the previous filling operation, a "backwards transport" along a path. For the goal of making a constructive model of univalent type theory, the choices of cube category and (trivial) cofibrations are not independent. If there are more maps in the cube category, then fewer cofibrations or trivial cofibrations may be required. For instance, if the cube category contains a reversal map $\mathbb{I} \to \mathbb{I}$, then all cubical sets (even non-Kan ones) admit reversal of paths, and it is not necessary to allow the $1/z$ trivial cofibration. On the other hand, if there are more maps in the cube category, then the uniformity conditions are harder to achieve (e.g. definition of the Kan operations from Bezem et al. (2018) are not uniform in diagonals).

The Kan operation that we use in this paper, Coquand's Kan operation for Cartesian cubical sets (Coquand, 2014*b*), generalizes the trivial cofibrations by allowing filling problems whose cap is located at $z = r$ (for an arbitrary map $r : \Psi \to \mathbb{I}$), not only $z = 0$ as in the previous filling operation:



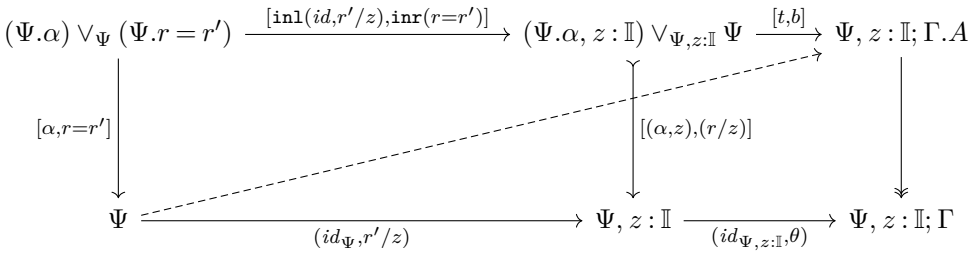
$$\frac{\begin{array}{l}\Psi \vdash r : \mathbb{I} \\ \Psi, z : \mathbb{I}; \Gamma \vdash A\ \mathtt{Type} \\ \Psi.\alpha, z : \mathbb{I}; \Gamma \vdash t : A \\ \Psi; \Gamma \vdash b : A\langle r/z\rangle \\ \Psi.\alpha; \Gamma \vdash t\langle r/z\rangle \equiv b : A\langle r/z\rangle\end{array}}{\begin{array}{l}\Psi, z : \mathbb{I}; \Gamma \vdash \mathtt{fill}_A^{z=r}(t, b) : A \\ \Psi, z : \mathbb{I}; \alpha; \Gamma \vdash \mathtt{fill}_A^{z=r}(t, b) \equiv t : A \\ \Psi; \Gamma \vdash (\mathtt{fill}_A^{z=r}(t, b))\langle r/z\rangle \equiv b : A\end{array}}$$

The motivation for this generalization is the definition of filling for Π-types: when filling in dimension z in $\Pi x{:}A.B$, it is natural to recursively use a contravariant filling in the domain type A, and the cap of this filling problem is at the filling dimension dimension z.

There is one final refinement of this Kan operation that is motivated by syntactic considerations: when an inference rule of a type theory would have a free variable in

the conclusion, it is typical to add an explicit substitution for that variable, so that substitution in general remains admissible (e.g. we turn $\Gamma, x : A, y : B \vdash (x, y) : A \times B$ into a rule that gives $\Gamma \vdash (a, b) : A \times B$ from $\Gamma \vdash a : A$ and $\Gamma \vdash b : B$). The filling rules as stated above have a free variable $z : \mathbb{I}$ in the conclusion standing for the filling dimension. Building in a substitution for z results in a *composition* operation instead of a filling one. The result of the composition operation should be thought of as "any of the faces in the z dimension of the filler." Because such faces includes a diagonal in a dimension z' in which the filling problem's data (the type, tube, and cap) is degenerate, we actually can obtain the entire filler this way as well. Formally, for any $r' : \Psi \to \mathbb{I}$, we obtain the above filler precomposed with the dimension substitution $\langle r'/z \rangle$:

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



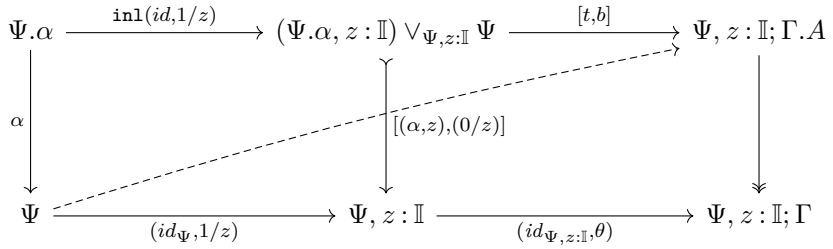
$$\frac{\begin{array}{c}\Psi \vdash r : \mathbb{I} \qquad \Psi \vdash r' : \mathbb{I} \qquad \Psi, z : \mathbb{I}; \Gamma \vdash A\ \mathtt{Type} \\ \Psi.\alpha, z : \mathbb{I}; \Gamma \vdash t : A \qquad \Psi; \Gamma \vdash b : A\langle r/z \rangle \qquad \Psi; \alpha; \Gamma \vdash t\langle r/z \rangle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

Note that the constraint $\mathtt{fill}\langle r/z \rangle \equiv b$ on fillers has been transformed into the constraint that $\mathtt{com}_A^{z:r\to r'}\ (\alpha \mapsto z.t)\ (b) \equiv b$ when $r = r'$. (This may be vacuous, e.g., when r is 0 and r' is 1.) In the diagram, the pushout-of-pullback in the top left encodes both the "restricts to t on α" and the "restricts to b on $r = r'$" constraints (and compatibility on both). Because composition transports the cap $b : A\langle r/x \rangle$ to an element of $A\langle r'/x \rangle$ (while possibly also attaching some tube faces), we refer to r as the "source" of the composition problem and r' as the "target."

The rule above appears in Brunerie and Licata (2014) (but with an explicit description of tubes rather than using formulas α); variations on this rule appear also in Angiuli et al. (2016); Angiuli and Harper (2017). Coquand (2014*b*); Brunerie and Licata (2014); Angiuli et al. (2016); Angiuli and Harper (2017) show that Π, Σ, path, and some higher inductive base types are closed under this Kan operation (satisfying the $r = r'$ constraint). However, the strict $r = r'$ constraint was an obstacle to closure of these models under univalent universes.

In the De Morgan setting Cohen et al. (2018) adopt a special case of the above Kan composition operation in which the source and target are fixed to 0 and 1 respectively

(see Sattler (2017) for a semantic analysis):



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

This specialization has the key advantage that the $r = r'$ constraint disappears, because $0 = 1$ never holds. Surprisingly, many operations that follow immediately from Definition 1 are derivable indirectly from the restricted $0 \to 1$ composition in the presence of connections and reversals.[j] In particular, the filler (i.e., the above rule with occurrences of 1 replaced by a fresh $\Psi \vdash z' : \mathbb{I}$) is definable as

$$\mathtt{com}_{z.A\langle(z' \wedge z)/z\rangle}(\alpha \vee (z' = 0) \mapsto z.[t\langle(z' \wedge z)/z\rangle, b])(b).$$

Definition 1 can be encoded in the De Morgan model (see Section 3.4), but the encoding transforms the $r = r'$ constraint of Diagonal Kan composition into an instance of the regularity condition that proved problematic in that model (as mentioned in Section 1.1). Regularity states that that a composition problem where all of the paths are the identity is the identity, i.e. $\mathtt{com}_{z.A}(\alpha \mapsto z.t)(b) \equiv b$ when z does not occur in A or in t. Thus, for several years it was an open problem whether univalent and Diagonal Kan universes could be defined for the Cartesian cube category.

1.3 Technical Contributions

The main new ingredient needed to show univalent and Diagonal Kan universes can be defined, discovered first in the computational setting by Angiuli et al. (2017*b*), and studied in a proof theory and cubical sets model in this paper, is to choose the cofibrations α to include the diagonal map $\mathbb{I} \to \mathbb{I} \times \mathbb{I}$ of $\mathbb{C}$. Geometrically, this corresponds to attaching faces on the diagonal of an open box. Syntactically, this corresponds to adding the formula $r = r'$ to the collection of generating cofibrations, for any two terms $\Psi \vdash r : \mathbb{I}$ and $\Psi \vdash r' : \mathbb{I}$. In a classical setting diagonal maps are cofibrations in Cisinski model structures, where the cofibrations are all monomorphisms. In a constructive setting there is an obligation that

[j] In the De Morgan model reversals are needed to derive "composition from 1" from "composition from 0," but the model also works (Orton and Pitts, 2016) in a cube category with only connections (and diagonals, faces, and degeneracies) by explicitly adding a second composition operation $1 \to 0$—i.e. taking $1/z$ in addition to $0/z$ as a trivial cofibration, but in both cases fixing the target as well as the source. The minor differences between this "connections model" and the De Morgan model are not relevant to this paper.

cofibrations be decidable (in a sense that will be made precise below), but this holds for pullbacks of the diagonal because equality of maps in the cube category is decidable.

Our Kan operation differs from the prior work (Awodey, 2018*b*) on constructing models of type theory in Cartesian cubical sets by using the generating trivial cofibrations of Definition 1 rather than only the endpoint inclusions $0/z$ and $1/z$, by taking the diagonal $\mathbb{I} \to \mathbb{I} \times \mathbb{I}$ as a generating cofibration, and by not including a regularity constraint on composition. We leave a formal comparison of the two models to future work—even though the *generating* (trivial) cofibrations are different, the full class of (trivial) cofibrations, defined as those maps with the left lifting property with respect to the fibrations, could still be the same. Our model provides univalent universes but does not support regularity, while Awodey's supports regularity but does not (at the time of this writing) include univalent universes. It remains open whether one can achieve regularity and univalent universes simultaneously—this would provide a constructive model of univalent type theory where Martin-Löf identity types can be interpreted as the path type, rather than via a different construction (Cohen et al., 2018; Cavallo and Harper, 2019).

In Section 2 we present a cubical type theory based on the Cartesian cube category, with Π, Σ, path, identity, natural number, boolean, suspension, glue, and universe types. For each type, we give a judgemental equality rule defining that type former's Kan operation in terms of the Kan operations of its component types.

Orton and Pitts (2016); Birkedal et al. (2018); Orton and Pitts (2018) 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 is well-suited to mechanization: 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). Notably Orton and Pitts (2016) mechanize much of the Cohen et al. (2018) model in this style. A later extension allows the internal description of universes (Licata et al., 2018), so that one can prove that they are fibrant and univalent.

In Section 3 we describe a (mechanically verified) constructive model in Cartesian cubical sets using the Orton–Pitts method. We have formalized the definition of diagonal Kan composition for glue, Π, Σ, path, identity, natural number, boolean, suspension, and universe types, and shown that the universe is univalent.[k] Our mechanization postulates the definitions of the 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 of Orton and Pitts (2016), we replace the axioms for connections on the interval with the axiom that the diagonal 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 et al., 2018). From these assumptions, our mechanization verifies all of the details of the Kan composition operations for these types, e.g. checking that the constructions in Section 2.11 type check and have the correct boundaries.

The mechanized internal language proof can be interpreted in presheaf toposes. In any presheaf topos let Ω_{dec} be the presheaf of decidable sieves (Orton and Pitts, 2016, Definition 6.2): at each Ψ, $\Omega_{dec}(\Psi)$ is the set of sieves on Ψ (precomposition-closed subsets of $\hom_C(-, \Psi)$) with the property that for a given map $\rho : \Psi' \to_C \Psi$ it is decidable whether

[k] Our Agda formalization can be found at https://dlicata.wescreates.wesleyan.edu/pubs/abcfhl/agda/ABCFHL-MSCS.html, and type checks using Agda version 2.6.1.

ρ is in the sieve.[1] Ω_{dec} is a subobject of the subobject classifier Ω. Then the internal language construction implies:

Theorem. *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}, which is closed under $=_{\mathbb{I}}$, $\vee$ and $\forall x : \mathbb{I}.-$. Then there is (for each size level i) a universe U_i classifying those semantic type families of size i equipped with a diagonal Kan composition structure (Definition 1) for generating cofibrations classified by Cof. U_i is closed under semantic Π, Σ, path, and glue types, and is itself Kan (U_{i+1} has a code for U_i). If $\hat{C}$ has cubical sets corresponding to boolean, natural number, and suspension types, then U_i is closed under those; if Cof is closed under $\wedge$, then U_i is closed under identity types as well.*

We can instantiate this theorem with the Cartesian cube category as C, and as *Cof* one of a number of possibilities, ranging from a minimal collection closed under only $\vee$ and $=_{\mathbb{I}}$ with $\forall$ defined by quantifier elimination, to a maximal one consisting of Ω_{dec} itself. In Section 3 we prove that the axioms used in our formalization hold in Cartesian cubical sets, using an argument similar to that of Orton and Pitts (2016); Licata et al. (2018). We briefly sketch the the interpretation of the syntax in this model, noting that the definitions of the Kan operations in the syntax and internal language model follow each other line by line. We can also instantiate this theorem with the De Morgan cube category, producing a second model in De Morgan cubical sets with a different notion of generating cofibrations/trivial cofibrations than Cohen et al. (2018). The two Kan operations are interderivable if we have both *both* connections (and reversal) and diagonal cofibrations.

The features of our Kan operation, assumptions about the interval, and closure conditions for cofibrations are used as follows:

- 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. This makes essential use of the generalized trivial cofibrations that we consider, because interchanging a filler results in a composite from a variable. The de Morgan model (Cohen et al., 2018) uses essentially the same definition (specialized to composition from 0 to 1), but both of these definitions are quite different than the proof of fibrancy of Π-types in simplicial sets (Kapulkin and Lumsdaine, 2021).
- Composition for Σ types uses the fact that Kan filling is derivable from composition.
- Composition for path types use the fact that the cofibrations include $x = 0$ and $x = 1$ and are closed under $\vee$.
- Composition for strict base types (natural numbers, booleans) use the connectedness axiom.
- Composition for higher inductive types uses a reduction of composition to coercion and homogeneous composition, which in turn uses "homogenization," another instance of our generalized trivial cofibrations.
- Constructing glue types uses the strictification axiom, which in a constructive model requires cofibrations to be decidable sieves.
- 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.

[1] This not the same as a proposition being decidable ($\phi \vee \neg\phi$) in the internal logic of the topos.

Readers who wish to learn how to *use* Cartesian cubical type theory can consult the `redtt` library at `https://github.com/RedPRL/redtt` for a variety of examples; Bentzen (2019), which derives the 1-groupoid laws, weak identity elimination, and the Eckmann–Hilton argument with our Kan operation; Angiuli (2019, Chapter 3), which derives weak identity elimination and discusses regularity; and the tutorials of Harper and Angiuli (2018); Harper (2018). Note that Section 2.16 (concerning identity types) implies that our type theory is an extension of standard ("book") homotopy type theory, so all constructions (for which the required inductive and higher inductive types exist) in (The Univalent Foundations Program, Institute for Advanced Study, 2013) can be interpreted.

Since the original draft of this article in December 2017, there have been have been many developments in cubical type theory, some building on the results presented here; we discuss these further in Section 4.

Acknowledgments We are very grateful to the late Vladimir Voevodsky for his vision and leadership, creating a rich research area that has captivated all of us over the past 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, Jon Sterling, and Todd Wilson for many helpful discussions and insights over the years leading to this work. Finally, we thank the anonymous referees for their helpful feedback on this work and paper.

This material is based on research sponsored by The United States Air Force Research Laboratory under agreement numbers FA9550-15-1-0053, FA9550-16-1-0292, and FA9550-21-0009 (Tristan Nguyen, program manager). 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

Cubical type theory extends the judgements of standard Martin-Löf type theory with interval variables and dimension formulas, represented by the following syntactic classes:

- Ψ is a *dimension context* representing an object Ψ in the cube category $\mathbb{C}$.
- $\Psi \vdash r : \mathbb{I}$ is a *dimension term* representing a map $\Psi \to \mathbb{I}$ in $\mathbb{C}$.
- $\Psi \vdash \phi\ \mathsf{formula}$ is a *dimension formula* representing a subobject of Ψ.
- $\Psi \vdash \alpha\ \mathsf{cofib}$ is a *cofibration*, a special kind of dimension formula used in Kan operations.
- $\phi \vdash_\Psi \alpha$ is the implication ordering on subobjects of Ψ, for $\Psi \vdash \phi\ \mathsf{formula}$ and $\Psi \vdash \alpha\ \mathsf{cofib}$.
- $\Psi; \phi \vdash \Gamma\ \mathsf{ctx}$ is a (ordinary) *context* relative to a dimension context Ψ and dimension formula $\Psi \vdash \phi\ \mathsf{formula}$.
- $\Psi; \phi; \Gamma \vdash A\ \mathtt{Type}_i$ is a type relative to a dimension context Ψ, a dimension formula $\Psi \vdash \phi\ \mathsf{formula}$, and a context $\Psi; \phi \vdash \Gamma\ \mathsf{ctx}$. We suppress levels i throughout, but formally types are stratified by size.
- $\Psi; \phi; \Gamma \vdash a : A$ is a term of a type $\Psi; \phi; \Gamma \vdash A\ \mathtt{Type}_i$, relative to a dimension context Ψ, a dimension formula $\Psi \vdash \phi\ \mathsf{formula}$, and a context $\Psi; \phi \vdash \Gamma\ \mathsf{ctx}$.

We also have equality judgements as follows, which are interpreted by (exact/strict) equality in models:

- Equality of dimension terms $\Psi; \phi \vdash r \equiv r' : \mathbb{I}$, assuming $\Psi \vdash r : \mathbb{I}$ and $\Psi \vdash r' : \mathbb{I}$ and $\Psi \vdash \phi\ \mathsf{formula}$.
- Equality of cofibrations $\Psi; \phi \vdash \alpha \equiv \alpha'\ \mathsf{cofib}$, assuming $\Psi \vdash \phi\ \mathsf{formula}$ and $\Psi \vdash \alpha\ \mathsf{cofib}$ and $\Psi \vdash \alpha'\ \mathsf{cofib}$.
- Equality of types $\Psi; \phi; \Gamma \vdash A \equiv A'\ \mathtt{Type}$, assuming $\Psi; \phi; \Gamma \vdash A\ \mathtt{Type}$ and $\Psi; \phi; \Gamma \vdash A'\ \mathtt{Type}$.
- Equality of terms $\Psi; \phi; \Gamma \vdash a \equiv a' : A$, assuming $\Psi; \phi; \Gamma \vdash a : A$ and $\Psi; \phi; \Gamma \vdash a' : A$.

We do not define equality judgements for contexts Ψ or ϕ or Γ, nor do we define proof terms for the (proof-irrelevant) subobject ordering $\alpha \vdash_\phi$.

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

2.2 Contexts

We have three kinds of contexts: dimension contexts Ψ, dimension formula contexts ϕ (geometrically representing subshapes of cubes), and ordinary contexts Γ of term variables. Dimension contexts are non-dependent; ϕ depends on Ψ but ϕ is not internally dependent.

$$\Psi ::= \cdot \mid \Psi, x : \mathbb{I}$$
$$\phi ::= \cdot \mid \phi, \alpha$$
$$\Gamma ::= \cdot \mid \Gamma, x : A$$

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

$$\frac{}{\Psi \vdash \cdot\ \mathsf{formula}} \qquad \frac{\Psi \vdash \phi\ \mathsf{formula} \quad \Psi \vdash \alpha\ \mathsf{cofib}}{\Psi \vdash \phi, \alpha\ \mathsf{formula}}$$

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

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 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—all rules treat dimension variables as placeholders, and do not, for example,

inspect whether a term is a variable, or whether two variables are different. Thus, we have silent weakening and exchange (where the dotted line indicates an admissible rule):

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

and substitution, with its usual composition law:

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

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

Semantically, a dimension context Ψ represents an object in the cube category $\mathbb{C}$ (or that object's Yoneda embedding in $\hat{\mathbb{C}}$). A dimension term $\Psi \vdash r : \mathbb{I}$ is a map $\Psi \to \mathbb{I}$ in $\mathbb{C}$. Because all objects of $\mathbb{C}$ 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.

The judgemental equality for interval terms is an equivalence relation:

$$\frac{}{\Psi; \phi \vdash r \equiv r : \mathbb{I}} \qquad \frac{\Psi; \phi \vdash r' \equiv r : \mathbb{I}}{\Psi; \phi \vdash r \equiv r' : \mathbb{I}} \qquad \frac{\Psi; \phi \vdash r \equiv r' : \mathbb{I} \quad \Psi; \phi \vdash r' \equiv r'' : \mathbb{I}}{\Psi; \phi \vdash r \equiv r'' : \mathbb{I}}$$

The only other way to obtain a judgemental equality of interval terms is via an equality reflection axiom relating to the formula ϕ, described shortly.

For cofibrations, types, and terms (i.e. the judgements that depend on interval variables and have a corresponding judgemental equality), the following functionality principles are admissible, stating that substituting equal dimension terms gives equals:

$$\frac{\Psi, x : \mathbb{I}, \Psi' \vdash \alpha\ \mathsf{cofib} \quad \Psi; \phi \vdash r \equiv r' : \mathbb{I}}{\Psi, x : \mathbb{I}, \Psi'; \phi, \phi'\langle r/x\rangle \vdash \alpha\langle r/x\rangle \equiv \alpha\langle r'/x\rangle\ \mathsf{cofib}}$$

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

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

2.4 Cofibrations

We use a syntactic notion of cofibration, in the style introduced by Cohen et al. (2018). Our presentation follows Riehl and Shulman (2017), and our discussion of their semantics follows Orton and Pitts (2016); Sattler (2017).

Our type theory has a notion of *dimension formula* $\Psi \vdash \phi\ \mathsf{formula}$. Thinking of Ψ as the representable presheaf $[\Psi] := \hom_{\mathbb{C}}(-, \Psi)$, we can regard $\Psi \vdash \phi\ \mathsf{formula}$ in several equivalent ways:

(1) A map $[\Psi] \to \Omega$ in $\hat{\mathbb{C}}$, where Ω is the subobject classifier of the presheaf topos $\hat{\mathbb{C}}$.
(2) A sieve on Ψ: a set of maps into Ψ closed under precomposition by all maps in $\mathbb{C}$. In a presheaf topos $\Omega(\Psi)$ is the set of sieves on Ψ, so this is the same as (1) by the Yoneda lemma.

(3) A subobject (subpresheaf) of $[\Psi]$, i.e., another presheaf $\Psi.\phi$ with a monic natural transformation $\phi : \Psi.\phi \hookrightarrow [\Psi]$.[m] The universal property of Ω expresses that this is equivalent to (1).
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 $[\Psi]$; such a morphism is necessarily monic. Moreover, because ϕ_1 is monic, any two such morphisms are equal, so the subobjects of Ψ form a poset $\mathbf{Sub}(\Psi)$.

Cofibrations are a designated subset of dimension formulas used to define Kan filling problems. We write $\mathbf{Cofibs}(\Psi)$ for the category of cofibrations into Ψ, which is a subposet of $\mathbf{Sub}(\Psi)$. Equivalently, one can define a cofibration classifier as a subobject of Ω. The judgement $\Psi \vdash \alpha\ \mathsf{cofib}$ is interpreted as an object $\alpha : \Psi.\alpha \hookrightarrow \Psi$ of $\mathbf{Cofibs}(\Psi)$ or as a map into the cofibration classifier. The rules for this judgement assert that cofibrations are closed under certain operations.

Rules for Dimension Formulas Syntactically, dimension formulas are contexts ϕ of cofibrations α. The empty formula context in Ψ corresponds to $id_\Psi : \Psi \hookrightarrow \Psi$, and context extension ϕ, α corresponds to the composite $\Psi.(\phi, \alpha) \hookrightarrow \Psi$ of the following pullback diagram:

$$\begin{array}{ccc} \Psi.(\phi,\alpha) & \dashrightarrow & \Psi.\alpha \\ \downarrow & \lrcorner & \downarrow \\ \Psi.\phi & \longrightarrow & \Psi \end{array}$$

(The composite is monic because monomorphisms are stable under pullback and closed under composition.) Equivalently, in any topos, this map is the product of ϕ and α in $\mathbf{Sub}(\Psi)$, or regarding ϕ, α as maps into Ω, the composition of (ϕ, α) with $\wedge : \Omega \times \Omega \to \Omega$. It is therefore sensible to treat ϕ, α as context extension.

In contrast with Cohen et al. (2018), we do not require these pullbacks to be cofibrations, except to construct identity types with an exact equality on `refl` (see Section 2.16).

Cofibrations Cofibrations are defined by the following rules:

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

These rules assert that the cofibrations into Ψ include equality on $\mathbb{I}$, and are closed under disjunction and universal quantification over $\mathbb{I}$. Substitution for $\Psi, x : \mathbb{I} \vdash \alpha\ \mathsf{cofib}$ states that cofibrations are closed under pullback along an arbitrary map in $\mathbb{C}$.

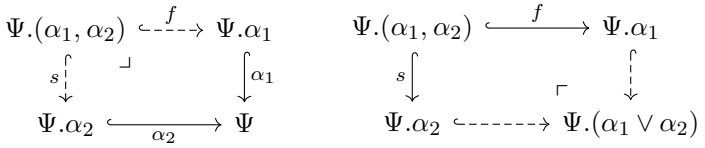
It is instructive to unpack these cofibrations as monomorphisms; $r = r'$ is (the Yoneda embedding of) a face, diagonal, or identity, or the unique map out of $\emptyset$ (the initial object of $\hat{\mathbb{C}}$), depending on the values of r and r':

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

[m] A subobject is technically an isomorphism class of such monomorphisms, identifying (A, h) and (B, k) when there is an isomorphism between A and B that sends h to k.

Admitting $x = y$ as a cofibration is the key ingredient used to define the diagonal Kan operation for the universe.

The subject of the second rule, $\alpha_1 \vee \alpha_2$, is the coproduct of cofibrations $\alpha_1 + \alpha_2$ in $\mathbf{Sub}(\Psi)$. In any topos this unfolds to the pushout of the pullback of the maps $\alpha_i : \Psi.\alpha_i \to \Psi$, which encodes an idea of *coherence* when α_1 and α_2 are both true:



We obtain $\Psi.(\alpha_1 \vee \alpha_2) \hookrightarrow \Psi$ by the universal property of the pushout applied to α_1 and α_2.

The third rule, concerning $\forall x : \mathbb{I}.\alpha$, takes as argument a cofibration $(\Psi, x : \mathbb{I}).\alpha \hookrightarrow (\Psi, x : \mathbb{I})$. In any topos the $\forall$ quantifier on subobjects $\forall : \mathbf{Sub}(\Psi, x : \mathbb{I}) \to \mathbf{Sub}(\Psi)$ is the right adjoint to pullback along the weakening $\Psi, x : \mathbb{I} \to \Psi$. Our rule asserts that $\forall$ preserves cofibrations.

The admissible weakening, exchange, and substitution principles for the dimension variable context Ψ in the judgement $\Psi \vdash \alpha\ \mathsf{cofib}$ express that cofibrations are closed under pullback along all maps in $\mathbb{C}$.

For most of our development, we do not need any non-trivial *equations* between cofibrations, such as $(\alpha \vee (0 = 1)) \equiv \alpha$ or $(\alpha_1 \vee \alpha_2) \equiv (\alpha_2 \vee \alpha_1)$, and require only that these pairs of cofibrations be interprovable. Semantically, these equations do hold by the "propositional univalence" of the subobject classifier in a topos (i.e., interprovable propositions are equal). We do, however, need these equations when constructing identity types from path types (see Section 2.16).

Subboundaries The judgement $\phi \vdash_\Psi \alpha$ states that there is a map (or entailment) $\phi \to \alpha$ in $\mathbf{Sub}(\Psi)$. Geometrically, this expresses that ϕ is a subspace of α, both as subspaces of Ψ. For example, $x = 0 \vdash_{x:\mathbb{I},y:\mathbb{I}} (x = 0) \vee (y = 1)$ is the inclusion of the left-hand side of a square into the left and top sides of a square, and $x = 0, y = 1 \vdash_{x:\mathbb{I},y:\mathbb{I}} x = 0$ is the inclusion of the top-left corner of a square into the left-hand side.

Because $\mathbf{Sub}(\Psi)$ is posetal, as previously discussed, we omit proof terms for this judgement in order to identify all derivations of $\phi \vdash_\Psi \alpha$.

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

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

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

$$\frac{\phi \vdash_\Psi 0 = 1}{\phi \vdash_\Psi \beta} \qquad \frac{\Psi \vdash r : \mathbb{I}}{\phi \vdash_\Psi (r = r)} \qquad \frac{\phi \vdash_\Psi (r = r')}{\Psi; \phi \vdash r \equiv r' : \mathbb{I}}$$

We present the rules in natural deduction style. The hypothesis rule and the admissible weakening and substitution rule express that context extension ψ, α is a product. We also have a conversion rule stating that definitional equality of cofibrations induces

an entailment. The rules for $\vee$ and $\forall$ are the usual natural deduction rules for these connectives—the rules for $\vee$ express the fact that $\alpha_1 \vee \alpha_2$ is a coproduct in $\mathbf{Sub}(\Psi)$, while the rules for $\forall$ express that it is right adjoint to weakening. For the equality cofibration, we stipulate that 0 is not equal to 1, that it is reflexive, and we give an equality reflection rule, stating that a proof of the proposition $r = r'$ entails a judgemental equality $r = r'$. (Because $\mathbf{Sub}(\Psi)$ is posetal, it is automatic that any proof of $r = r'$ is reflexive.)

Judgemental equality of cofibrations is a congruence (we do not explicitly write the rules making it reflexive, symmetric, and transitive, with congruence rules for $\vee$ and $\forall$) built from the judgemental equality of interval terms in the base case of the equality cofibration:

$$\frac{\Psi; \phi \vdash r \equiv r_1 : \mathbb{I} \quad \Psi; \phi \vdash r' \equiv r'_1 : \mathbb{I}}{\Psi; \phi \vdash (r = r') \equiv (r_1 = r'_1)\ \mathsf{cofib}}$$

Typical proofs of the equality cofibration use this rule together with equality reflection and conversion. For example, to prove $r = r' \vdash_\Psi r' = r$, by equality reflection we have $r \equiv r'$, and by symmetry we have $r' \equiv r$, so the cofibrations $r = r$ and $r' = r$ are equal, and the reflexivity proof of $r = r$ is also a proof of $r' = r$. Alternatively, one can use an admissible "transport" rule (taken as a defining rule in the tope logic of (Riehl and Shulman, 2017)):

$$\frac{\Psi, x : \mathbb{I} \vdash \alpha\ \mathsf{cofib} \qquad \phi \vdash_\Psi r = r' \qquad \phi \vdash_\Psi \alpha\langle r/x\rangle}{\phi \vdash_\Psi \alpha\langle r'/x\rangle}$$

Given the above premises, we have $r \equiv r'$ by equality reflection, and $\alpha\langle r/x\rangle \equiv \alpha\langle r'/x\rangle$ by the admissible functionality principle discussed previously; the conclusion follows from conversion.

Boundaries and partial elements The type and term formation judgements take place in context $\Psi; \phi; \Gamma \vdash J$. The $(\Psi; \phi)$ part can be understood as the domain of the monomorphism $\Psi.\phi \hookrightarrow [\Psi]$, or intuitively, "the subset of Ψ on which ϕ holds." Weakening, exchange, and substitution for cofibrations are admissible for type and term judgements:

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

Most of our cofibrations are left-invertible; we give left rules for the type and term formation judgements, though we could avoid duplication if we identified types and elements of a universe.

In an ambient context $\Psi; \phi$, a *partial element* of a type is a term $\Psi; \phi, \alpha \vdash a : A$ for some cofibration α. We will sometimes write $\alpha \vdash a : A$, leaving the ambient content implicit, allowing us to conveniently and concisely state many judgemental equality axioms as "$\alpha \vdash t \equiv t'$." Such rules should be understood to mean that, in a general $\Psi; \phi; \Gamma$, if t and t' are well-formed and moreover $\phi \vdash_\Psi \alpha$, the equation holds.

We axiomatize contradiction by the rules:

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

We axiomatize $\vee$ using its characterization as the pushout of a pullback:

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

$$\alpha \vee \beta \vdash t \equiv [\alpha \mapsto t, \beta \mapsto t] \qquad \alpha \vee \beta \vdash A \equiv [\alpha \mapsto A, \beta \mapsto A]$$

Equality reflection and the admissible functionality rules imply admissible congruence rules for the $r = r'$ cofibration (these are taken as defining rules in the tope logic of Riehl and Shulman (2017)):

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

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

When we have a nested disjunction $\alpha_1 \vee \alpha_2 \vee \alpha_3 \vee \ldots \vee \alpha_n$, we write a nested case with the same associativity as the cofibration as $[\alpha_1 \mapsto t_1, \ldots, \alpha_n \mapsto t_n]$.

2.5 Judgemental equality

Typing and term equality respect equality of types:

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

Judgemental equality is a congruence (reflexive, symmetric, and transitive):

$$\frac{\Psi;\phi;\Gamma \vdash A\ \mathtt{Type}}{\Psi;\phi;\Gamma \vdash A \equiv A\ \mathtt{Type}} \qquad \frac{\Psi;\phi;\Gamma \vdash A \equiv B\ \mathtt{Type}}{\Psi;\phi;\Gamma \vdash B \equiv A\ \mathtt{Type}}$$

$$\frac{\Psi;\phi;\Gamma \vdash A \equiv B\ \mathtt{Type} \quad \Psi;\phi;\Gamma \vdash B \equiv C\ \mathtt{Type}}{\Psi;\phi;\Gamma \vdash A \equiv C\ \mathtt{Type}}$$

$$\frac{\Psi;\phi;\Gamma \vdash a : A}{\Psi;\phi;\Gamma \vdash a \equiv a : A} \qquad \frac{\Psi;\phi;\Gamma \vdash a \equiv b : A}{\Psi;\phi;\Gamma \vdash b \equiv a : A} \qquad \frac{\Psi;\phi;\Gamma \vdash a \equiv b : A \quad \Psi;\phi;\Gamma \vdash b \equiv c : A}{\Psi;\phi;\Gamma \vdash a \equiv c : A}$$

We also require congruence rules for each type and term constructor, using the appropriate type or term or dimension term equality judgement for each subterm, though we do not explicitly write these below. From these, the following functionality principles are

admissible:

$$\frac{\Psi;\phi;\Gamma, x:A, \Gamma' \vdash B \equiv B' \;\mathtt{Type} \quad \Psi;\phi;\Gamma \vdash a \equiv a' : A}{\Psi;\phi;\Gamma, \Gamma'[a/x] \vdash B[a/x] \equiv B'[a'/x]\;\mathtt{Type}}$$

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

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

Our structural rules for terms are typical—variable usage is a defining rule, while weakening, exchange (assuming independence), and substitution are admissible:

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

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

Composition laws for substitution hold syntactically:

$$u[a/x] \equiv_\alpha u \qquad \text{when } x\#u$$
$$u[a/x][b/y] \equiv_\alpha u[b/y][a[b/y]/x]$$

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 operation

All types in our syntax are *Kan* (or *fibrant*), i.e., equipped with the following composition operation:

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

We read the composition 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 b." The boundary condition

on $r = r'$ states that this operation is the identity when the source and target agree; the boundary condition on α states that the composite agrees with the r' side of the input partial element $z.t$. The latter ensures, for example, that the left and right endpoints of the missing face of $\sqcup$ agree with the top-left and top-right corners of the input $\sqcup$.

The main challenge of cubical type theory is to define the $\mathtt{com}$ operation for each type former in terms of the $\mathtt{com}$ operations of its constituent types. We will make use of various lemmas and derived forms, presented below. Note that $\mathtt{com}$ respects judgemental equality in the type argument, and therefore judgementally equal types must be assigned the same Kan operation; this is the major challenge for glue types (equivalently, univalent universes).

Filling from composition Kan filling, as opposed to Kan composition, is the "whole cube" that extends the input data, instead of just the "missing side." We obtain filling by composing to a fresh variable, or geometrically, degenerating and composing to a diagonal. When the target of a composition is a variable z' that does not occur in $r, A, \alpha, z.t, b$, we will sometimes write:

$$\Psi, z' : \mathbb{I}; \phi; \Gamma \vdash \mathtt{fill}_A^{z:r\to z'}\ (\alpha \mapsto z.t)\ (b) := \mathtt{com}_A^{z:r\to z'}\ (\alpha \mapsto z.t)\ (b) : A\langle z'/z\rangle$$

This degenerates all components of the composition problem in a fresh direction z', and then takes the z'-diagonal (in this case just a renaming of variables) of the original filling direction.

One can easily verify the following boundary conditions, noting in the third case that $\langle z'/z\rangle$ acts as a renaming rather than a diagonal, because z' does not occur in t.

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

Extending partial elements in contractible types The usual definition of a *contractible* type in univalent foundations is that $\mathsf{Contractible}(A) := \Sigma x{:}A.\Pi y{:}A.\mathtt{Path}_A\ (x, y)$ is inhabited. Fibrant types that are contractible in this sense have the property that any partial element extends to a total element Cohen et al. (2018, Section 5.1). In model category theory terms, this says that fibrant contractible types have the right lifting property against cofibrations, i.e. that they can be seen as trivial fibrations. In more detail, given $c : \mathsf{Contractible}(A)$, a cofibration α, and a partial element $\alpha \vdash a : A$, we can define the following, where, following Cohen et al. (2018), the notation $b : A[\alpha \mapsto a]$ abbreviates the two judgements $a : A$ and $\alpha \vdash b \equiv a : A$.[n]

$$\mathsf{contr\_extend\_partial}(\alpha.a) := \mathtt{com}_{\overline{A}}^{\_:0\to 1}\ (\alpha \mapsto z.\mathtt{snd}(c)\ a\ z)\ (\mathtt{fst}(c)) : A[\alpha \mapsto a]$$

Here, $\mathtt{snd}(c)\ a$ has type $\mathtt{Path}_A\ (\mathtt{fst}(c), a)$, so composing the center of contraction $\mathtt{fst}(c)$ with that path on α yields a total element of A that is a on α. (See Section 2.10 for the rules for path types.) Indeed, the property "any partial element of A extends to a total element" is itself fibrant, a (-1)-type, and equivalent to $\mathsf{Contractible}(A)$, so it could be taken to be the *definition* of contractibility in a cubical type theory.

Voevodsky defines $f : T \to B$ to be an *equivalence* when $\Pi b{:}B.\mathsf{Contractible}(\mathsf{HFiber}(f, b))$, where $\mathsf{HFiber}(f, b) := \Sigma t{:}T.\mathtt{Path}_B\ (f(t), b)$ is the *homotopy fiber* of f at b. We therefore obtain the equiv operation of Cohen et al. (2018) as a corollary: if $f : A \to B$ is an equivalence, then for any $b : B$, any partial element of $\mathsf{HFiber}(f, b)$ extends to a total element.

[n] We write "_" for a bound dimension variable that does not appear in its scope.

Adjusting a composition structure by a partial composition structure The $\mathtt{com}$ operator assigns to each type a composition operation satisfying a number of conditions. However, a particular type may admit other terms besides $\mathtt{com}_A^{z:r\to r'}\,(\alpha \mapsto z.t)\,(b)$ that also satisfy the typing and equality rules of $\mathtt{com}$. We call any such term a *composition structure* for A (of which $\mathtt{com}$ is an example), and write com instead of $\mathtt{com}$ to (subtly) differentiate the two:

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

We write an inference rule with a double-line to indicate derivability—a double line is often used for invertibility, but here we imagine it as hiding a derivation between the two lines.

Given a composition structure com_A for $z:\mathbb{I} \vdash A\ \mathtt{Type}$, and under a cofibration β, a partial composition structure pcom_A for $z:\mathbb{I}; \beta \vdash A\ \mathtt{Type}$, we can define a (total) composition structure $\mathsf{adjust\_com}_A$ for $z:\mathbb{I} \vdash A\ \mathtt{Type}$ that agrees with pcom_A on β:

$$\mathsf{adjust\_com}_A^{z:r\to r'}(\alpha \mapsto z.t)(b) := \mathsf{com}_A^{z:r\to r'}[\alpha \mapsto z.t, \beta \mapsto z'.\mathsf{pcom}_A^{z:r\to z'}(\alpha \mapsto z.t)(b)](b)$$

Homogeneous composition and coercion Following the previous definition, we say that a *homogeneous composition (hcom) structure* on a type A is a term hcom_A satisfying:

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

An hcom structure is a restricted form of composition structure in which the type A is not allowed to depend on the filling direction.

A *coercion structure* on $z:\mathbb{I} \vdash A\ \mathtt{Type}$ is a composition structure with no α boundary constraint:

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

Every composition structure com_A gives rise to hcom and coercion structures on A:

$$\mathsf{hcom}_A^{r\to r'}\,(\alpha\mapsto z.t)\,(b) := \mathsf{com}_A^{z:r\to r'}\,(\alpha\mapsto z.t)\,(b)$$
$$\mathsf{coe}_A^{z:r\to r'}(b) := \mathsf{com}_A^{z:r\to r'}\,(0=1\mapsto z.\mathtt{abort})\,(b)$$

We write `hcom` and `coe` for the hcom and coercion structures obtained in this way from a canonical composition structure `com`.

Conversely, an hcom structure and a coercion structure give rise to a composition structure:

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

The operation $\mathsf{coe}_A^{z:z'\to r'}(t\langle z'/z\rangle)$ that occurs in the tube of this definition "moves" $t\langle z'/z\rangle$ from a diagonal to r'. For example, in the case where $r'=0$ and writing t' for $t\langle z'/z\rangle$, we have $z':\mathbb{I}\vdash t':A\langle z'/z\rangle$, i.e. t' is a heterogeneous path in the line $A\langle z'/z\rangle$. Then $z':\mathbb{I}\vdash\mathsf{coe}_A^{z:z'\to 0}(t'):A\langle 0/z\rangle$ is a homogeneous path in the fiber over 0, whose left endpoint is $\mathsf{coe}_A^{z:1\to 0}(t\langle 0/z\rangle)$ and whose right endpoint is $t\langle 1/z\rangle$. So this instance of composition is a *homogenization* operation that turns a heterogeneous path ("path over") into a homogeneous-path-with-a-`transport`. Heterogeneous composition uses this homogenization to move all of the pieces of the composition problem into the target fiber r', and then performs a homogeneous composition in that fiber.

This decomposition of composition as homogeneous composition and coercion was first used by Coquand (2015) to define composition for higher inductive types. Some sources take hcom and coercion as primitive instead of composition, including Angiuli et al. (2017*b*) and an implementation of the De Morgan model with regularity (Coquand, 2014*a*). Here, we adopt this decomposition only in certain types, such as higher inductive types. Recent work on higher inductives in the De Morgan model (Coquand et al., 2018) uses a similar decomposition, but coercion must be generalized to take a cofibration on which the coercion is constant—this is not needed in our setting because the empty-tube instance of the diagonal Kan operation already includes homogenization (because the source r can be a variable, as described above).

Weak coercion A *weak coercion structure* on $z:\mathbb{I}\vdash A\ \mathtt{Type}$ is similar to a coercion structure, except that on $r=r'$ it is only an identity function up to a path β_b:

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

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

Of course, any coercion structure (hence, any composition structure) gives rise to a weak coercion structure. More surprisingly, diagonal cofibrations (and an hcom structure) allow us to improve weak coercion structures to strict ones, a maneuver we will use in higher inductive types:

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

Therefore, if we have a weak coercion structure on $z : \mathbb{I} \vdash A\ \mathtt{Type}$ and an hcom structure for each fiber of A, we can obtain a composition structure on $z : \mathbb{I} \vdash A\ \mathtt{Type}$.

Strictly preserving homogeneous compositions Given $x : A \vdash f : B$, an hcom structure on A, and a composition structure on B, we say that f *strictly preserves hcoms* if

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

There is always a path between these equands; in higher inductive types it is a judgemental equality.

2.8 Σ types

The formation, introduction, elimination, β, and η rules for Σ types are the usual ones, but apply at any dimension; we can therefore consider pairs of not only points, but also paths, squares, etc.

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

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

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

Kan operation

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

The computation rule for composition generalizes the usual rule for `transport` at Σ-types: push into both components, the second over the first. In the second component we substitute $\mathtt{fill}^{z:r\to y}_A\ (\alpha \mapsto z.\mathtt{fst}(t))\ (\mathtt{fst}(b))$ into $B\langle y/z\rangle$ for a fresh y. This makes the second component type-check: under $\langle r/y\rangle$, the filler reduces to $\mathtt{fst}(b)$, which appears in the type of $\mathtt{snd}(b)$; on α, the filler reduces to $\mathtt{fst}(t)$, which occurs in the type of $\mathtt{snd}(t)$; and under $\langle r'/y\rangle$, the filler is the first component of the pair.

2.9 Π types

Once again the standard rules for Π types hold at any dimension.

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

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

$$(\lambda x.u)\, a \equiv u[a/x]$$

$$f \equiv \lambda\, x.\, f\, x$$

Kan operation

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

Here, note that $B\langle y/z\rangle[\mathtt{fill}^{z:r'\to y}_{A}\ (a)/x]$ under $\langle r/y\rangle$ agrees with $B\langle r/z\rangle[\mathtt{coe}^{z:r'\to r}_{A}(a)/x]$ (ensuring that the argument $b\,(\mathtt{coe}^{z:r'\to r}_{A}(a))$ has the correct type), and under $\langle r'/y\rangle$ agrees with $B\langle r'/z\rangle[a/x]$ (ensuring that the result type is correct).

In Section 1 we motivated Kan composition *to* an arbitrary r' as a natural way to close Kan filling under substitution. Our definition of $\mathtt{com}$ from r to r' for Π-types requires the *reverse* coercions and fillers, from r' to r. Therefore, for Π-types to be closed under our Kan operation—at least for the present definition of composition in Π-types—allowing the target of a composition problem to be an arbitrary r' requires us to allow the *source* also to be an arbitrary r'.

2.10 Path types

Paths are functions out of $\mathbb{I}$ with specified behavior on 0 and 1. The elimination rules for path types state that an element u can be turned back into a cube in A by instantiating it with a dimension r. When r is a dimension variable that does not occur in u, this operation simply chooses a name for the "hidden" dimension of the path type element. When r is a dimension variable that does occur, this operation takes a diagonal. When r is 0 or 1, $u\,r$ is equal to the element specified by u's type, ensuring that u connects the specified points. 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.

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

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

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

$$u\, 0 \equiv a_0$$

$$u\, 1 \equiv a_1$$

$$(\Lambda x.u)\, r \equiv u\langle r/x\rangle$$

$$u \equiv \Lambda x.(u\, x)$$

Kan operation

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

On the right-hand side, we extend the cofibration of `com` by $(x=0)\vee(x=1)$ to ensure that the body of the Λ has the correct endpoints. Recall that the commas inside $[-]$ construct a partial element under a disjunction of cofibrations; the new faces are compatible with $\alpha\mapsto z.t\,x$ because t's path type ensures that $t\,0$ and $t\,1$ equal a_0 and a_1 respectively.

2.11 Glue (equivalence extension) types

The "glue" or equivalence extension type $\mathtt{Glue}\,(\alpha\mapsto(T,f))\,(B)$, introduced by Coquand (2014*a*); Cohen et al. (2018), is a type that is equal to the partial type T on α, given a function $\alpha\vdash f:T\to B$ mapping T into the total type B. The formation, introduction, elimination, β, and η rules of `Glue` types hold for arbitrary such f, but `Glue` types are only Kan when f is an equivalence.

Because all types in our syntax are Kan, we restrict the formation rule of `Glue` types to equivalences $f:A\simeq B$, defined as functions $f:A\to B$ whose homotopy fibers are contractible, as in Section 2.7. (We implicitly coerce equivalences to functions $A\to B$ when necessary.) In the semantics of Section 3 we define `Glue` types for any f, but the universe of Kan types only includes codes for `Glue` types of equivalences. In the formalization we show that all such `Glue` types support a *homogeneous composition* structure.

The rules below essentially state that $\mathtt{Glue}\,(\alpha\mapsto(T,f))\,(B)$ consists of pairs of $b:B$ and $\alpha\vdash t:T$ such that $\alpha\vdash f(t)\equiv b:B$, with a first projection `unglue`. However, `Glue` types are stricter than Σ types: under α, the `Glue` type is judgementally equal to T, its elements restrict to t, and the first projection restricts to f.

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

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

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

$$\mathtt{unglue}(\mathtt{glue}\,(\alpha\mapsto t)\,(b))\equiv b$$

$$g\equiv \mathtt{glue}\,(\alpha\mapsto g)\,(\mathtt{unglue}(g))$$

The Kan operation for glue types is quite complicated, and we obtain it in multiple steps.

Weak glue introduction The arguments of $\texttt{glue}\,(\alpha \mapsto t)\,(b)$ are $b : B$ and $\alpha \vdash t : T$ such that $\alpha \vdash f(t) \equiv b : B$. If we weaken this judgemental equality to a path, then the pair of t and such a path is a (partial) element of the homotopy fiber of f at b, $\mathsf{HFiber}(f, b)$. Using homogeneous composition in B, we can derive a weak version of $\texttt{glue}$ that takes an element of $\mathsf{HFiber}(f, b)$:

$$\frac{b : B \quad \alpha \vdash h : \mathsf{HFiber}(f, b)}{\begin{array}{c}\mathsf{wglue}'\,(\alpha \mapsto h)\,b : \texttt{Glue}\,(\alpha \mapsto (T, f))\,(B) \\ := \texttt{glue}\,(\alpha \mapsto \texttt{fst}(h))\,(\texttt{com}_B^{:1\to 0}\,(\alpha \mapsto x.\texttt{snd}(h)\,x)\,(b))\end{array}}$$

We adjust b by $\texttt{snd}(h) : \texttt{Path}\_.B\,(f(\texttt{fst}(h)), b)$, obtaining an element of B that is $f(\texttt{fst}(h))$ on α.

In the Kan operation for glue types we will need a slightly stricter version of wglue'. First, note that if $g : \texttt{Glue}\,(\alpha \mapsto (T, f))\,(B)$, then under α, g is in the fiber (and hence, homotopy fiber) of $f : T \to B$ over $\texttt{unglue}(g)$:

$$\alpha \vdash \mathsf{glue\_to\_fiber}(g) : \mathsf{HFiber}(f, \texttt{unglue}(g)) := (g, \Lambda\_.\texttt{unglue}(g))$$

The first component of the pair is well-typed because $\alpha \vdash \texttt{Glue}\,(\alpha \mapsto (T, f))\,(B) \equiv T\ \texttt{Type}$, and the second is well-typed because $\alpha \vdash \texttt{unglue}(g : \texttt{Glue}\,(\alpha \mapsto (T, f))\,(B)) \equiv f(g) : B$.

Now, suppose we have a partial element $\beta \vdash g : \texttt{Glue}\,(\alpha \mapsto (T, f))\,(B)$ that agrees in the appropriate sense with both $b : B$ (i.e., under $\texttt{unglue}$) and $\alpha \vdash h : \mathsf{HFiber}(f, b)$ (i.e., under $\mathsf{glue\_to\_fiber}$). Then we can obtain a total element of the glue type that restricts on β to g:

$$\frac{\begin{array}{c}\beta \vdash g : \texttt{Glue}\,(\alpha \mapsto (T, f))\,(B) \\ b : B[\beta \mapsto \texttt{unglue}(g)] \\ \alpha \vdash h : \mathsf{HFiber}(f, b)[\beta \mapsto \mathsf{glue\_to\_fiber}(g)]\end{array}}{\begin{array}{c}\mathsf{wglue}\,(\alpha \mapsto h)\,(b)\,(\beta \mapsto g) : \texttt{Glue}\,(\alpha \mapsto (T, f))\,(B)[\beta \mapsto g] \\ := \texttt{glue}\,(\alpha \mapsto \texttt{fst}(h))\,(\texttt{com}_B^{:1\to 0}\,[\alpha \mapsto x.\texttt{snd}(h)\,x, \beta \mapsto \_.\texttt{unglue}(g)]\,(b))\end{array}}$$

Incoherent composition for glue types In the next step we define a candidate composition structure for glue types satisfying:

$$\frac{z : \mathbb{I} \vdash G := \texttt{Glue}\,(\alpha \mapsto (T, f))\,(B)\,\texttt{Type} \quad s : \mathbb{I} \quad s' : \mathbb{I} \quad z : \mathbb{I}, \beta \vdash u : G \quad v : G\langle s/z\rangle[\beta \mapsto u\langle s/z\rangle]}{\mathsf{icom}_G^{z:s\to s'}(\beta \mapsto z.u)v : G\langle s'/z\rangle[\beta \mapsto u\langle s'/z\rangle, s = s' \mapsto v]}$$

However, this structure will be "incoherent" in the sense that it will not restrict to $\texttt{com}_T$ under α. As a result, we will *not* be able to define $\texttt{com}_G \equiv \mathsf{icom}_G$ directly: under α, the left-hand side will equal $\texttt{com}_T$ but the right-hand side will not.

We begin by constructing the filler and composite of $\texttt{unglue}(u)$ and $\texttt{unglue}(v)$ in B, which form a filling problem in B because u, v form one in G by assumption:

$$\begin{array}{rll} w : \mathbb{I} \vdash b_{\mathit{fill}} & := \texttt{fill}_B^{z:s\to w}\,(\beta \mapsto z.\texttt{unglue}(u))\,(\texttt{unglue}(v)) & : B\langle w/z\rangle \\ b' & := b_{\mathit{fill}}\langle s'/w\rangle & : B\langle s'/z\rangle \end{array}$$

Next, we define a partial element under $\alpha\langle s'/z\rangle$, the cofibration under which $G\langle s'/z\rangle$ (the type of icom) restricts to $T\langle s'/z\rangle$. The data of the glue type G includes a function:

$$\alpha\langle s'/z\rangle \vdash f\langle s'/z\rangle : T\langle s'/z\rangle \to B\langle s'/z\rangle$$

as well as a proof f_{eq} that this function is an equivalence:

$$\alpha\langle s'/z\rangle \vdash f_{eq} : \Pi b{:}B\langle s'/z\rangle.\mathsf{Contractible}(\mathsf{HFiber}(f\langle s'/z\rangle, b))$$

Applying f_{eq} to the composite b' defined above, and using contr_extend_partial (Section 2.7), we see that any partial element of $\mathsf{HFiber}(f\langle s'/z\rangle, b')$ extends to a total one. We will apply this fact to the following partial element:

$$\beta \vee s = s' \vdash [\beta \mapsto \mathsf{glue\_to\_fiber}(u\langle s'/z\rangle), s = s' \mapsto \mathsf{glue\_to\_fiber}(v)] : \mathsf{HFiber}(f\langle s'/z\rangle, b')$$

This is well-typed because $\beta \vdash b' \equiv \mathtt{unglue}(u\langle s'/z\rangle)$ and $s = s' \vdash b' \equiv \mathtt{unglue}(v)$ by the definition of b' above, and $\beta, s = s' \vdash u\langle s'/z\rangle \equiv u\langle s/z\rangle \equiv v$ by assumption. (Note that we are using the fact that general equality on $\mathbb{I}$, and hence $s = s'$, is a cofibration.) Summing up, we have:

$$\alpha\langle s'/z\rangle \vdash c : \mathsf{HFiber}(f\langle s'/z\rangle, b')[\beta \mapsto \mathsf{glue\_to\_fiber}(u\langle s'/z\rangle), s = s' \mapsto \mathsf{glue\_to\_fiber}(v)]$$
$$c := \mathsf{contr\_extend\_partial}(f_{eq}(b'))[\beta \mapsto \mathsf{glue\_to\_fiber}(u\langle s'/z\rangle), s = s' \mapsto \mathsf{glue\_to\_fiber}(v)]$$

Finally, we construct an element of $G\langle s'/z\rangle = \mathtt{Glue}\,(\alpha\langle s'/z\rangle \mapsto (T\langle s'/z\rangle, f\langle s'/z\rangle))\,(B\langle s'/z\rangle)$ using wglue, which requires an element of $B\langle s'/z\rangle$ (for which we choose b'), an element of $\alpha\langle s'/z\rangle \vdash \mathsf{HFiber}(f\langle s'/z\rangle, b')$ (we choose c), a cofibration (we choose $\beta \vee s = s'$), and an element of $\beta \vee s = s' \vdash G\langle s'/z\rangle$ (we choose $[\beta \mapsto u\langle s'/z\rangle, s = s' \mapsto v]$). We must check two equations: first, that $\beta \vee (s = s') \vdash b' \equiv \mathtt{unglue}([\beta \mapsto u\langle s'/z\rangle, s = s' \mapsto v])$, which holds by the definition of b', and that $\alpha\langle s'/z\rangle, \beta \vee (s = s') \vdash c \equiv \mathsf{glue\_to\_fiber}([\beta \mapsto u\langle s'/z\rangle, s = s' \mapsto v])$, which holds by the definition of c. Thus, overall, we have

$$\mathsf{icom}_G^{z:s\to s'}\,(\beta \mapsto z.u)v := \mathsf{wglue}\,(\alpha\langle s'/z\rangle \mapsto c)\,(b')\,[\beta \mapsto u\langle s'/z\rangle, s = s' \mapsto v]$$

Aligning Christian Sattler and Ian Orton (independently) analyzed the algorithm for composition in glue types given by Cohen et al. (2018), and realized that the use of the $\forall$ cofibration could be isolated in a single "aligning" step that fixes an incoherent composition operation for glue types so that it restricts appropriately to composition in T on α.

We use the same method here, applying adjust_com (Section 2.7) to a total and a partial composition structure for $z : \mathbb{I} \vdash G := \mathtt{Glue}\,(\alpha \mapsto (T, f))\,(B)\,\mathtt{Type}$. For the total composition structure we choose icom_G, for the cofibration we choose $\forall z : \mathbb{I}.\alpha$, and for the partial composition structure under $\forall z : \mathbb{I}.\alpha$ we choose $\mathtt{com}_T$. The latter is a partial composition structure for G because $\forall z : \mathbb{I}.\alpha \vdash G \equiv T$, using the fact that $\forall z : \mathbb{I}.\alpha \vdash_{z:\mathbb{I}} \alpha$. Concretely, we therefore obtain:

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

This "aligned" definition satisfies:

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

and therefore the same equation holds under α (because $\forall z : \mathbb{I}.\alpha$ implies α), as required by the equation $\alpha \vdash \mathtt{Glue}\,(\alpha \mapsto (T, f))\,(B) \equiv T$.

Comparison with CCHM The CCHM algorithm inlines the wglue lemma, the adjust_com lemma, and the aligning step; otherwise, the main difference is that here we use the diagonal cofibration $s = s'$ in contr_extend_partial to "remember" that the result should be v on $s = s'$.

2.12 Universes

Depending on technical details of the intended semantics, we could define universes à la Tarski or à la Russell; we elide the $El(-)$ for notational simplicity. Universes are specified by the usual rules:

$$\frac{\Psi;\phi;\Gamma\vdash A\,\mathtt{Type}_i}{\Psi;\phi;\Gamma\vdash A:\mathtt{U}_i}\qquad\frac{\Psi;\phi;\Gamma\vdash A:\mathtt{U}_i}{\Psi;\phi;\Gamma\vdash A\,\mathtt{Type}_i}\qquad\frac{}{\Psi;\phi;\Gamma\vdash \mathtt{U}_i:\mathtt{U}_{i+1}}$$

Suppose $\mathtt{U}$ is a universe of Kan types that is closed under glue types. For $\mathtt{U}$ to itself be Kan, we need a (homogeneous) composition structure on it, which is to say that whenever $B:\mathtt{U}$ and $z,\alpha\vdash T:\mathtt{U}$ with $\alpha\vdash T\langle r/z\rangle\equiv B$, we need a Kan type:

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

Here, following Cohen et al. (2018), we construct such a type by converting paths in the universe to equivalences and using the fact that `Glue` types are Kan for equivalences. We define:

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

where e is a proof that $\mathtt{coe}_T^{z:r'\to r}(-)$ is an equivalence and e' is a proof that $\lambda x.x$ is an equivalence. We can define e' in a standard fashion using contractibility of singletons, which follows from composition and filling; we obtain e by transporting e':

$$e:=\mathtt{coe}^{z':r'\to r}_{\mathsf{isEquiv}(\lambda x.\mathtt{coe}_T^{z:r'\to z'}(x))}(e'),$$

which agrees with e' under $r=r'$ as required.

This type satisfies the required boundary equations:

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

Following Cohen et al. (2018), we can construct a proof of the univalence axiom for $\mathtt{U}$, using the fact that $\mathtt{U}$ is Kan and closed under glue types. An equivalence $f:A\simeq B$ gives rise to a path in $\mathtt{U}$ as follows:

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

This glue type restricts to A under $x=0$ and B under $x=1$ as required; both follow from the equation $\alpha\vdash\mathtt{Glue}\,(\alpha\mapsto(T,f))\,(B)\equiv T$ we obtained by alignment in Section 2.11.

By an internal argument of Licata (2016) using ideas of Egbert Rijke and Martín Escardó, the full univalence axiom $(A\simeq B)\simeq\mathtt{Path\_}_{\mathtt{U}}\,(A,B)$ follows from $\mathtt{ua}(f)$ and a term:

$$\mathtt{ua}_\beta(f):\mathtt{Path\_}_{A\to B}\,(\mathtt{com}^{z:0\to1}_{\mathtt{ua}(f)\,z}\,[]\,(-),f)$$

stating that transport along $\mathtt{ua}(f)$ applies f. In our formalization we construct $\mathtt{ua}_\beta$ by inspecting the above definition of composition in glue types.

An alternative construction of universes, given by Angiuli et al. (2017*b*), is to define $\mathtt{com}_{\mathtt{U}}$ as a type former directly (analogous to `Glue`), and separately to define a "V type" implementing the $\mathtt{ua}(-)$ operation above. This essentially factors `Glue` into two smaller type formers, operating on types and equivalences respectively.

2.13 Strict booleans

The introduction and elimination rules for booleans are the usual ones:

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

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

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

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

$$\frac{\Psi, x : \mathbb{I};\phi;\Gamma \vdash u : \mathtt{bool} \qquad \Psi \vdash r, r' : \mathbb{I}}{\Psi;\phi;\Gamma \vdash u\langle r/x\rangle \equiv u\langle r'/x\rangle : \mathtt{bool}}$$

A consequence of this rule is that $p : \mathtt{Path}_{\_.\mathtt{bool}}\,(b_0, b_1)$ entails judgemental equalities $b_0 \equiv b_1$ (by taking $p\ x$ for u and $0, 1$ for r, r') and $p \equiv \Lambda\_.b_0$ (taking $0, x$ for r, r'). Therefore, this rule justifies the following definition (for `bool` and any other closed base type with decidable equality):

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

If one wants judgemental equality to be decidable, one should instead omit the above "equality reflection" rule, and, following Cohen et al. (2018), instead define the Kan operation as:

$$\mathtt{com}^{\_:r\to r'}_{\mathtt{bool}}\ (\alpha \mapsto \_.\mathtt{true})\ (\mathtt{true}) \equiv \mathtt{true}$$
$$\mathtt{com}^{\_:r\to r'}_{\mathtt{bool}}\ (\alpha \mapsto \_.\mathtt{false})\ (\mathtt{false}) \equiv \mathtt{false}$$

A third option is to treat `bool` as if it were a higher inductive type, in which case its values are not only `true` and `false` but also formal homogeneous compositions (see Section 2.15). In this paper such booleans would contain non-standard points in the empty context, such as $\mathtt{com}^{\_:0\to 1}_{\mathtt{bool}}\ (0 = 1 \mapsto \_.\mathtt{abort})\ (\mathtt{true})$. Angiuli et al. (2017*b*) prohibit false cofibrations in the composition operation (and therefore rule out such spurious compositions); in exchange they must handle $\forall x : \mathbb{I}.-$ specially, because it can produce false cofibrations. In their system, higher inductives have only point constructors as elements in the empty context (in this case, just `true` and `false`).

2.14 Strict natural numbers

Again, the introduction and elimination rules are standard:

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

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

$$\mathbb{N}\text{-}\mathtt{elim}_{x.C}(\mathtt{succ}(m), v_0, n.r.v_1) \equiv v_1[m/n][\mathbb{N}\text{-}\mathtt{elim}_{x.C}(m, v_0, n.r.v_1)/r]$$

Analogously to `bool`, one way to define the Kan operation is via equality reflection:

$$\frac{\Psi, x : \mathbb{I};\phi;\Gamma \vdash u : \mathtt{nat} \quad \Psi \vdash r, r' : \mathbb{I}}{\Psi;\phi;\Gamma \vdash u\langle r/x\rangle \equiv u\langle r'/x\rangle : \mathtt{nat}} \qquad \mathtt{com}^{:r\to r'}_{\mathtt{nat}}\ (\alpha \mapsto z.t)\ (b) \equiv b$$

2.15 Suspension types

We consider here only suspensions as an illustration of higher inductive types, noting that suspensions and booleans suffice to construct all spheres. Higher inductive types are treated systematically for diagonal Kan composition by Cavallo and Harper (2019) and Cavallo (2021).

The introduction forms are the usual higher inductive constructors, plus a free homogeneous composition structure `vhcom`; the eliminator computes strictly on all of these:

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

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

$$\frac{\begin{array}{l}\Psi \vdash r, r' : \mathbb{I} \\ \Psi;\phi;\Gamma \vdash A\ \mathtt{Type} \\ \Psi, z : \mathbb{I};\phi,\alpha;\Gamma \vdash t : \mathbf{\Sigma} A \\ \Psi;\phi;\Gamma \vdash b : \mathbf{\Sigma} A \\ \Psi;\phi,\alpha;\Gamma \vdash t\langle r/z\rangle \equiv b : \mathbf{\Sigma} A\end{array}}{\begin{array}{l}\Psi;\phi;\Gamma \vdash \mathtt{vhcom}^{r\to r'}_{\mathbf{\Sigma} A}\ (\alpha \mapsto z.t)\ (b) : \mathbf{\Sigma} A \\ r = r' \vdash \mathtt{vhcom}^{r\to r'}_{\mathbf{\Sigma} A}\ (\alpha \mapsto z.t)\ (b) \equiv b \\ \alpha \vdash \mathtt{vhcom}^{r\to r'}_{\mathbf{\Sigma} A}\ (\alpha \mapsto z.t)\ (b) \equiv t\langle r'/z\rangle\end{array}} \qquad \frac{\begin{array}{l}\Psi;\phi;\Gamma, a : \mathbf{\Sigma} A \vdash C\ \mathtt{Type} \\ \Psi;\phi;\Gamma \vdash u : \mathbf{\Sigma} A \\ \Psi;\phi;\Gamma \vdash v_0 : C[\mathtt{north}/a] \\ \Psi;\phi;\Gamma \vdash v_1 : C[\mathtt{south}/a] \\ \Psi, z : \mathbb{I};\phi;\Gamma, b : A \vdash v_2 : C[\mathtt{merid}_z(b)/a] \\ \Psi;\phi;\Gamma, b : A \vdash v_2\langle 0/z\rangle \equiv v_0 : C[\mathtt{north}/a] \\ \Psi;\phi;\Gamma, b : A \vdash v_2\langle 1/z\rangle \equiv v_1 : C[\mathtt{south}/a]\end{array}}{\Psi;\phi;\Gamma \vdash \mathbf{\Sigma}^{a.C}_{\mathrm{elim}}(v_0; v_1; z.b.v_2; u) : A[u/a]}$$

$$\begin{aligned}
\mathbf{\Sigma}^{a.C}_{\mathrm{elim}}(v_0; v_1; z.b.v_2; \mathtt{north}) &\equiv v_0 \\
\mathbf{\Sigma}^{a.C}_{\mathrm{elim}}(v_0; v_1; z.b.v_2; \mathtt{south}) &\equiv v_1 \\
\mathbf{\Sigma}^{a.C}_{\mathrm{elim}}(v_0; v_1; z.b.v_2; \mathtt{merid}_r(u)) &\equiv v_2\langle r/z\rangle[u/b] \\
\mathbf{\Sigma}^{a.C}_{\mathrm{elim}}(v_0; v_1; z.b.v_2; \mathtt{vhcom}^{r\to r'}_{\mathbf{\Sigma}A}\ (\alpha \mapsto z.t)\ (b)) & \\
&\equiv \mathtt{com}^{z':r\to r'}_{C[\mathtt{vhcom}^{r\to z'}_{\mathbf{\Sigma}A}\ (\alpha\mapsto z.t)\ (b)/a]}\ (\alpha \mapsto z.E(t))\ (E(b))
\end{aligned}$$

In the last equation we write $E(x)$ for $\mathbf{\Sigma}^{a.C}_{\mathrm{elim}}(v_0; v_1; z.b.v_2; x)$; it states that E strictly preserves hcoms, sending $\mathtt{vhcom}_{\mathbf{\Sigma}A}$ to $\mathtt{com}_C$ judgementally.

In the semantics we define $\mathbf{\Sigma}A$ as the least cubical set satisfying the introduction rules above; the elimination rule then holds for any type family fibrant over $\mathbf{\Sigma}A$ (i.e., for which $a : \mathbf{\Sigma}A \vdash C\ \mathtt{Type}$ has a composition operation) (Coquand, 2015). To define a composition structure on $\mathbf{\Sigma}A$, we decompose composition into homogeneous composition and coercion, and coercion into weak coercion and homogeneous composition (Section 2.7). We take $\mathtt{vhcom}^{r\to r'}_{\mathbf{\Sigma}A}\ (\alpha \mapsto z.t)\ (b)$ as the homogeneous composition structure.

To obtain a weak coercion structure, we first define the following operation:

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

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

Note that we cannot define $\mathtt{wcoe}_{\mathbf{\Sigma}A}$ as an instance of $\mathbf{\Sigma}_{\mathrm{elim}}$, because we do not know that $\mathbf{\Sigma}A$ is fibrant—that is what we are trying to show! We can, however, define it by external induction on the elements of $\mathbf{\Sigma}A$.

To show that $\mathtt{wcoe}^{z:r\to r'}_{\mathbf{\Sigma}A}\ (-)$ is a weak coercion structure, we must show that it is homotopic to the identity when $r = r'$, again by induction on the elements of $\mathbf{\Sigma}A$:

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

$$\begin{aligned}
\eta(\mathtt{north}) &\equiv \Lambda_.\mathtt{north} \\
\eta(\mathtt{south}) &\equiv \Lambda_.\mathtt{south} \\
\eta(\mathtt{merid}_s(u)) &\equiv \Lambda_.(\mathtt{merid}_s(u))
\end{aligned}$$

$$\begin{aligned}
&\eta(\mathtt{vhcom}^{s\to s'}_{\mathbf{\Sigma}A}\ (\alpha \mapsto z'.t)\ (b)) \\
&\equiv \Lambda x.\mathtt{vhcom}^{s\to s'}_{\mathbf{\Sigma}A}\ [\alpha \mapsto z'.\eta(t)\ x, x = 0 \mapsto z'.\mathtt{wcoe}^{z:r\to r}_{\mathbf{\Sigma}A}\ (t), x = 1 \mapsto z'.t]\ (\eta(b)\ x)
\end{aligned}$$

2.16 Identity types

Cohen et al. (2018) use an idea of Andrew Swan to construct identity types (that is, with a judgemental equality for the J eliminator on $\mathtt{refl}$) from path types (which model J only up to a path). The same construction applies in our model, though the definition of J is a bit more complex. In both cases, J follows from transport and contractibility of

singleton types; in CCHM, the latter is an immediate consequence of connections, but here it requires Kan composition. Cavallo and Harper (2019) give an alternate construction of identity types that takes certain transports in the identity type to be constructors, analogously to taking certain homogeneous compositions to be values for higher inductive types.

For this section only we add the following cofibration rules:

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

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

The first rule is "propositional univalence" (interprovable cofibrations are equal); the others state that cofibrations are closed under conjunction.

An element of the identity type is a path together with a cofibration remembering where the path is judgementally constant:

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

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

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

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

In the above we write $\mathtt{refl}_a$ for $(0 = 0, \Lambda\_.a)$. We reduce `J` to transport and singleton contractibility in the standard way; the computation rule for `J` follows from the fact that transport judgementally sends `refl` to the identity function, and singleton contractibility judgementally sends `refl` to `refl`. We define transport as:

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

To see that the `com` is well-typed, note that $z : \mathbb{I}; \alpha \vdash b : C[p\,z/x]$ because $\alpha \vdash p \equiv \Lambda\_.a_0$. By inspection, `transport` on $(0 = 0, \Lambda\_.a_0)$ sends b to b.

Write $S(a_1)$ for the singleton type $\Sigma x{:}A.\mathtt{Id}_A(x, a_1)$. We define singleton contractibility as:

$$\frac{a_0 : A \quad (\alpha, p) : \mathtt{Id}_A(a_0, a_1)}{\begin{array}{c}\mathtt{scontr}(a_0, (\alpha, p)) := (\alpha, \Lambda x.(s\langle 0/y\rangle, ((x = 0 \vee \alpha), \Lambda y.s))) : \mathtt{Id}_{S(a_1)}((a_1, \mathtt{refl}_{a_1}), (a_0, (\alpha, p))) \\ \text{where } x : \mathbb{I}, y : \mathbb{I} \vdash s := \mathtt{hcom}_A^{1\to y}\ [x = 0 \mapsto \_.a_1,\ x = 1 \mapsto y.p\, y,\ \alpha \mapsto \_.a_1]\ (a_1)\end{array}}$$

Because `scontr` outputs the same cofibration α as the input path, it sends `refl` to `refl`. Thus, `J` on `refl` cancels judgementally.

Kan operation

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

It is straightforward to see that the second component is constant on $(\alpha \wedge \alpha_t\langle r'/z\rangle) \vee (r = r' \wedge \alpha_b)$: under $\alpha \wedge \alpha_t\langle r'/z\rangle$, we have $\alpha \vdash \mathtt{com} \equiv t\langle r'/z\rangle$ and $z : \mathbb{I}; \alpha_t \vdash t \equiv \Lambda\_.a_0$, and under $r = r' \wedge \alpha_b$ we have $r = r' \vdash \mathtt{com} \equiv b$ and $\alpha_b \vdash b \equiv \Lambda\_.a_0$.

We must also check that this element of the identity type has the correct boundary under α and under $r = r'$. Under $r = r'$, it must equal (α_b, b), which requires the following cofibrations to be equal:

$$r = r' \vdash ((\alpha \wedge \alpha_t\langle r'/z\rangle) \vee (r = r' \wedge \alpha_b)) \equiv \alpha_b$$

This is a consequence of propositional univalence for cofibrations. The reverse implication is clear (by proving the right disjunct); the forward implication follows from $\alpha \vdash \alpha_t\langle r/z\rangle \equiv \alpha_b$, which we have from the original composition problem. Under α, the argument is analogous: we need

$$\alpha \vdash ((\alpha \wedge \alpha_t\langle r'/z\rangle) \vee (r = r' \wedge \alpha_b)) \equiv \alpha_t\langle r'/z\rangle,$$

which follows from propositional univalence and the fact that $\alpha, r = r' \vdash \alpha_t\langle r'/z\rangle \equiv \alpha_b$.

3. Semantics

The syntactic presentation in Section 2 is intended to admit a model in Cartesian cubical sets and other presheaf categories with suitable structure. We construct our model using the internal language of the topos of Cartesian cubical sets, a technique first investigated by Orton and Pitts (2016, 2018); Birkedal et al. (2018). Following Orton and Pitts (2016, 2018), we use the Agda proof assistant to simulate the internal language of Cartesian cubical sets, using the *modal* or *flat* features of Agda (developed by Andrea Vezzosi) to internally describe fibrant universes (Licata et al., 2018).

3.1 Overview of the formalization

We begin by describing the main definitions in our Agda formalization, explain what is proved, and give a sample proof. The formalization is browsable in HTML format at https://dlicata.wescreates.wesleyan.edu/pubs/abcfhl/agda/ABCFHL-MSCS.html and can be downloaded from https://dlicata.wescreates.wesleyan.edu/pubs/abcfhl/abcfhl.tar.gz.

3.1.1 Assumptions

We make Agda into a pseudo-extensional type theory by postulating function extensionality (Lib.λ=) and a (-1)-truncated/squashed disjunction (Proposition.Or) written $\alpha 1 \vee \alpha 2$. We define Proposition.Proposition as the type of strict propositions, types P paired with proofs that $(x\,y : P) \to x = y$.[o] (The formalization uses universe polymorphism, so there are propositions in each universe.)

Next, we make several postulates that are a subset of those of Orton and Pitts (2016), except for Cofibs.isCofib=, which allows diagonal cofibrations. In Interval, we postulate an interval type

```
𝕀 : Set
'0 : 𝕀
'1 : 𝕀
```

which is non-trivial

```
iabort : '0 = '1 → ⊥ {lzero}
```

and connected

```
iconnected : {ℓ : Level} (P : 𝕀 → Proposition {ℓ})
  → ((i : 𝕀) → (fst (P i) ∨ (fst (P i) → ⊥ {ℓ})))
  → (((i : 𝕀) → fst (P i)) ∨ ((i : 𝕀) → (fst (P i)) → ⊥ {ℓ}))
```

In Cofibs, we postulate a type of cofibrations closed under $\forall$, $\vee$, and $=$ of elements of $\mathbb{I}$:

```
isCofib : Set → Set
isCofib⊥ : isCofib ⊥
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 $t : \alpha \to A$ is a partial element of A, we write (Cofibs.extends)

```
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.

In Strictify, we add an axiom stating 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.

```
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) ]
```

Following Licata et al. (2018), we also postulate that the interval is *tiny*, i.e., that exponentiation by the interval has a right adjoint (illustrated in Figure 1 of that paper). In our formalization, tininess justifies our construction of universes of fibrant types in Section 3.1.5, which follows Theorem 5.2 of Licata et al. (2018).

Finally, our construction of identity types (Section 2.16) requires three additional postulates in Id: any two proofs that a type is a cofibration are equal, cofibrations are closed under $\wedge$, and interprovable cofibrations are equal.

```
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')
```

[o] At the time of this writing, there is a bug in some PDF readers on MacOS that prevents the links directly to Agda identifiers in files from working. To work around this bug, please either use Acrobat Reader, or click the link and then change %23 to #.

```
Cofib-propositional-univalence : ∀ {α α'} {{cα : Cofib α}} {{cα' : Cofib α'}}
  → (α → α') → (α' → α) → α = α'
```

3.1.2 Kan operation

We represent the diagonal Kan operation in much the same way that Orton and Pitts (2016) represent the CCHM Kan operation. In Kan.hasCom, we define composition structures for an $\mathbb{I}$-indexed family (of Sets, which here is a universe of cubical sets), following the rule in Section 2.7:

```
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) ]
```

The ⇒ stands for transport along the Agda equality type, which would be silent in an actual extensional type theory; here, `(fst b) : A r` but the partial element must have type `(r = r') → A r'`.

Then, for a type `A` dependent on Γ, a composition structure relative to Γ (Kan.relCom) 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 = Γ} A = (p : 𝕀 → Γ) → hasCom (A ∘ p)
```

This matches the diagrams in Section 1, because `p` may have free variables besides the bound $\mathbb{I}$.

3.1.3 Path types

Path types (Path.PathO) are defined as pairs of functions from the interval with proofs of the boundary constraints.

```
PathO : {ℓ : Level} (A : 𝕀 → Set ℓ) (a0 : A '0) (a1 : A '1) → Set ℓ
PathO A a0 a1 = Σ [p : (x : 𝕀) → A x] ((p '0 = a0) × (p '1 = a1))
```

This validates the formation, introduction, and elimination rules of Section 2.10, except that in Agda we have explicit proofs of the boundary conditions, which in our rules appear only in derivations, not terms.

3.1.4 Glue types

The cubical sets underlying glue types are parametrized by a cofibration α, a partial type `T` defined on α, a total type `B`, and a partial function `f` (under α) from `T` to `B`:

```
{ℓ : Level} (α : Set)
{{_ : Cofib α}}
(T : α → Set ℓ)
(B : Set ℓ)
(f : (u : α) → T u → B)
```

Following Orton and Pitts (2016), we define glue types as a strictification of pairs of partial elements `t` of `T` with elements of `B` that agree with `f t` on α:

```
Σ [t : (pα : α) → T pα] (B [ α ↦ (λ pα → f pα (t pα)) ])
```

In Glue, we define the following terms (each parametrized additionally by the above α, `T`, `B`, `f`), which correspond to the rules in Section 2.11:

```
Glue : Set ℓ
Glue-α : (u : α) → Glue α T B f = T u
glue : (top : ((u : α) → T u))
  (base : B [ α ↦ (λ u → f u (top u)) ])
  → Glue α T B f
glue-α : (top : ((u : α) → T u))
  (base : B [ α ↦ (λ u → f u (top u)) ])
  (u : α) → coe (Glue-α α T B f u) (glue α T B f top base) = top u
unglue : Glue α T B f → B
unglue-α : (g : Glue α T B f) → (u : α) → f u (coe (Glue-α _ _ _ _ u) g) = unglue g
Glueβ : (top : ((u : α) → T u))
  (base : B [ α ↦ (λ u → f u (top u)) ])
  → unglue (glue α T B f top base) = fst base
Glueη : (g : Glue α T B f)
  → g = (glue α T B f (λ u → coe (Glue-α α T B f u) g) (unglue g , unglue-α g))
```

Here, `coe : A = B → A → B` is transport along the Agda equality type.

3.1.5 Universes

When we unfold the internal language of Cartesian cubical sets, Agda's universes `Set ℓ` correspond to Hofmann–Streicher universes of cubical sets (Hofmann and Streicher, 1997). To model the cubical type theory of Section 2 in which types are *Kan* cubical sets, we must find universes `U` that classify cubical sets equipped with Kan operations. That is, maps `A : Γ → U` should correspond to types fibrant over Γ, or families `A' : Γ → Set` equipped with a choice of Kan structure of type `relCom A'`.

We cannot describe such a universe directly in pure Agda for reasons described by Licata et al. (2018, Theorem 3.1), but we *can* describe it using a recently-added *modal* extension of Agda, implemented by Andrea Vezzosi, that allows one to talk about "closed" or "external" elements of a type.[p] This extension allows us to write `f : (@♭ x : A) → B (x)` for a "function with a flat domain"; the force of this restriction is that `f` can only be applied to terms that have no free variables (or only flat variables free).

When we unfold the internal language, a (closed) Agda type `A` denotes a cubical set, while `♭ A` denotes (the discrete cubical set on) the set of 0-cells of `A`. Licata et al. (2018) describe the flat variable mechanism in more detail; the main thing to understand for this paper is that Agda checks that the following axioms obey the modal typing discipline.

In universe.Universe, we axiomatize universes of fibrant types as follows (parametrized by `{@♭ ℓ : Level}`), using an unpacked version of Licata et al. (2018, (16)):

```
U : Set (ℓ2 ⊔ lsuc ℓ)
El : U → Set ℓ
comEl : relCom El
code : {@♭ ℓ1 : Level} (@♭ Γ : Set ℓ1) (@♭ A : Γ → Set ℓ) (@♭ comA : relCom A)
  → Γ → U
code-El : {@♭ ℓ1 : Level} {@♭ Γ : Set ℓ1} {@♭ A : Γ → Set ℓ} {@♭ comA : relCom A}
  → (x : Γ) → El ((code Γ A comA) x) = A x
comEl-β : {@♭ ℓ1 : Level} {@♭ Γ : Set ℓ1} {@♭ A : Γ → Set ℓ} {@♭ comA : relCom A}
  → (comEl' (code Γ A comA)) = comA
code-η : {@♭ ℓ1 : Level} {@♭ Γ : Set ℓ1} (@♭ A : Γ → U)
  → A = code Γ (El' A) (comEl' A)
```

[p] The code accompanying Licata et al. (2018) uses a prototype implementation of this extension in a branch of Agda named agda-flat, but as of Agda version 2.6.1 the modal features are included in the main branch.

The first line says that U is an Agda type (of universe level at least 2—because the Kan operation quantifies over cofibrations, which are sets of level zero, we represent types by Agda universes of level at least 1). The second gives the decoding function El that interprets each element of U as an Agda type. The third says that El is Kan; because being Kan is closed under precomposition, this implies that for any function A : Γ → U, El ∘ A is Kan (universe.Universe.U.comEl').

```
comEl' : {l1 : Level} {Γ : Set l1} (A : Γ → U) → relCom (El ∘ A)
```

The fourth is the introduction rule for the universe, which pairs a type family with a composition structure. The modal annotations here ensure that code is only applied to "closed" families. That is, A is not allowed to have other free variables besides Γ; if it did, the resulting map into the universe would allow proving that it is Kan relative to them, which is not justified by the introduction rule. Finally, we have β and η rules: El and comEl' project the arguments of code, and modal maps into the universe are determined by their type family and composition structure.

These universes are open-ended, in the sense that code includes in U any type for which a Kan operation can be defined—it is not necessary to fix a collection of types when defining the universe.

3.1.6 Suspensions

To define suspensions (Susp), we postulate the underlying cubical set (introduction, elimination, and computation rules), and define a composition structure following the argument in Section 2.15. The postulates are:

```
Susp : (A : Set ℓ) → Set ℓ
north : {A : Set ℓ} → Susp A
south : {A : Set ℓ} → Susp A
merid : {A : Set ℓ} → A → 𝕀 → (Susp A)
merid0 : {A : Set ℓ} (x : A) → merid x '0 = north
merid1 : {A : Set ℓ} (x : A) → merid x '1 = south
fcomSusp : {A : Set ℓ} → hasCom (λ _ → Susp A)
Susp-elim : {ℓ ℓ' : Level} {A : Set ℓ}
            (C : Susp A → Set ℓ')
            (comC : relCom C)
            (n : C north)
            (s : C south)
            ((a : A) → PathO (λ x → C (merid a x)) n s)
            (x : Susp A) → C x
```

This asserts a suspension type Susp.Susp with point constructors north and south, path constructor merid with specified boundary, and a homogeneous composition constructor fcomSusp, which freely adds homogeneous composites in Susp A (represented by asserting that the constant family Susp A is Kan). Finally, we postulate an elimination rule, which eliminates into any type family that is Kan over Susp A (along with computation rules, which we omit here). Other higher inductive types follow a similar pattern.

3.1.7 Main theorem

Theorem 1. *Each universe* U{ℓ+1} *has codes for* Π *(universe.Pi),* Σ *(universe.Sigma),* Path *(universe.Path),* Id *(universe.Id and Id),* Nat *(universe.Nat),* Bool *(universe.Bool),* Glue *(universe.Glue),* U{l} *(universe.U), and* Susp *(universe.Susp) types, and is univalent (universe.Univalence). That is, for all* {𝒪 ℓ1 ℓ2 : Level}*, we have the following:*

```
Πcode     : {Γ : Set ℓ1} (A : Γ → U {ℓ2}) (B : Σ (El o A) → U {ℓ2}) → (Γ → U {ℓ2})
El-Πcode  : {Γ : Set ℓ1} (A : Γ → U {ℓ2}) (B : Σ (El o A) → U {ℓ2})
  → (θ : Γ) → El (Πcode A B θ) = ((x : El (A θ)) → El (B (θ , x)))

Σcode     : {Γ : Set ℓ1} (A : Γ → U {ℓ2}) (B : Σ (El o A) → U {ℓ2}) → (Γ → U {ℓ2})
El-Σcode  : {Γ : Set ℓ1} (A : Γ → U {ℓ2}) (B : Σ (El o A) → U {ℓ2})
  → (θ : Γ) → El (Σcode A B θ) = (Σ [x : El (A θ) ] El (B (θ , x)))

Path-code     : {Γ : Set ℓ1} (A : Γ × 𝕀 → U {ℓ2})
  (a0 : (θ : Γ) → El (A (θ , '0))) (a1 : (θ : Γ) → El (A (θ , '1))) → Γ → U {ℓ2}
El-Path-code : {Γ : Set ℓ1} (A : Γ × 𝕀 → U {ℓ2})
  (a0 : (x : Γ) → El (A (x , '0))) (a1 : (x : Γ) → El (A (x , '1)))
  (θ : Γ) → El (Path-code A a0 a1 θ) = PathO (λ x → El (A (θ , x))) (a0 θ) (a1 θ)

Id-code :  {Γ : Set ℓ1} (A : Γ → U {ℓ2})
  (a0 : (θ : Γ) → El (A θ)) (a1 : (θ : Γ) → El (A θ)) → Γ → U {lsuc lzero ⊔ ℓ2}
refl      :  (A : Set ℓ1) (a0 : A) → Id A a0 a0
J         :  (A : U {ℓ1}) → (a0 : El A)
  (C : (Σ [a : El A] Id (El A) a a0) → U {ℓ2})
  (c : El (C (a0 , refl (El A) a0)))
  → Σ [f : (a1 : El A) (p : Id (El A) a1 a0) → El (C (a1 , p)) ]
  f a0 (refl (El A) a0) = c

Nat-code     : {Γ : Set ℓ2} → (Γ → U {ℓ1})
Nat-code-El : {Γ : Set ℓ2} → (θ : Γ) → El (Nat-code θ) = Nat

Bool-code     : {Γ : Set ℓ2} → (Γ → U {ℓ1})
Bool-code-El : {Γ : Set ℓ2} → (θ : Γ) → El (Bool-code θ) = Bool

Glue-code' : {Γ : Set ℓ1} (α : Γ → Set) (cα : (θ : Γ) → Cofib (α θ))
  (T : (θ : Γ) → α θ → U {ℓ2}) (B : Γ → U {ℓ2})
  (f : (θ : Γ) (pα : α θ) → El (T θ pα) → El (B θ))
  (feq : (θ : Γ) (pα : α θ) → isEquivFill _ _ (f θ pα))
  → Γ → U {ℓ2}
Glue-code-El : {Γ : Set ℓ1} (α : Γ → Set) (cα : (θ : Γ) → Cofib (α θ))
  (T : (θ : Γ) → α θ → U {ℓ2}) (B : Γ → U {ℓ2})
  (f : (θ : Γ) (pα : α θ) → El (T θ pα) → El (B θ))
  (feq : (θ : Γ) (pα : α θ) → isEquivFill _ _ (f θ pα))
  (θ : Γ) → El (Glue-code' α cα T B f feq θ) =
  Glue (α θ) {{cα θ}} (El o (T θ)) (El (B θ)) (f θ)

U-code     :  {𝒪 ℓ : Level} → U {ℓ2 ⊔ lsuc ℓ}
U-code-El :  {𝒪 ℓ : Level} → El (U-code {ℓ}) = U

ua    : {A B : U {ℓ1}} (e : Equiv (El A) (El B)) → Path U A B
uaβ : {A B : U {ℓ1}} (e : Equiv (El A) (El B)) (a : El A)
  → Path _ (coePathU (ua {_} {A} {B} e) a) (fst e a)

Susp-code     : {Γ : Set ℓ1} → (Γ → U {ℓ2}) → Γ → U {ℓ2}
Susp-code-El : {Γ : Set ℓ1} → (A : Γ → U {ℓ2}) (θ : Γ) → El (Susp-code A θ) = Susp (El (A θ))
```

For each type, we show that there is a code in the universe(s) U of fibrant types which satisfies the correct type formation rule, and whose elements are the corresponding Agda type (which implies that the necessary cubical set constructs exist; for example, Π decodes to Agda's Π-types, and so has λ and application). Id types satisfy refl and J with an exact equality for J on refl. ua states that an equivalence can be turned into a path in the universe, while uaβ gives a path between coercing along ua and the input equivalence.

3.1.8 Proof

The main work in establishing this theorem is to define the Kan operation of each type. These definitions in our formalization follow exactly the same steps as the equations

for each type given in Section 2. We show the definition for Π types (universe.Pi) as an example.

The formation rule of a Π-type takes an element of the universe and a family over that type:

```
ΠIData : (@♭ ℓ : Level) → Set _
ΠIData ℓ = Σ [A : U {ℓ} ] (El A → U {ℓ})
```

This determines the obvious Π-type:

```
Π-from-data : ∀ {@♭ l : Level} → ΠIData ℓ → Set _
Π-from-data (A , B) = (x : El A) → El (B x)
```

In Section 2.9, we showed that Π-from-data is Kan with the following term:

$$\lambda a'.\mathtt{com}^{y:r\to r'}_{B\langle y/z\rangle[\mathtt{fill}^{z:r'\to y}_A\ (a')/x]}\ (\alpha \mapsto y.t\langle y/z\rangle\ (\mathtt{fill}^{z:r'\to y}_A\ (a'))) (b\ (\mathtt{coe}^{z:r'\to r}_A(a')))$$

This definition makes crucial use of the Diagonal Kan operation's generalized source—the source of the contravariant coercion/filling in A is r', which might be a variable, not an endpoint 0 or 1.

In Agda, this is rendered as

```
comΠ : ∀ {@♭ ℓ : Level} → relCom {Γ = (ΠIData ℓ)} Π-from-data
comΠ AB r r' α t b =
  (λ a' → ... (fst (forward a'))) ,
  ... ,
  ... where
  A = λ z → fst (AB z)
  B = λ za → snd (AB (fst za)) (snd za)

  fillback : (a' : _) (y : 𝕀) → _
  fillback a' y = coeU (λ z → A z) r' y a'

  forward : (a' : _) → _
  forward a' = comEl (λ y → B (y , (fst (fillback a' y)))) r r'
                     α (λ y pα → t y pα (fst (fillback a' y)))
                     (fst b (fst (fillback a' r)) , ...)
```

fillback corresponds to the fill in the type of the B composition and in the tube, and fillback a' r to $\mathtt{coe}^{z:r'\to r}_A(a')$. The ellipses elide Agda proof terms for propositional equalities, which witness that the restrictions of the output are correct (on α, it restricts to t r', and on r=r', it restricts to b), and the boundary condition required by the composition in B in forward.

From this, we define a "universal" code for Π-types, i.e., a type constructor $A : U, B : A \to U \vdash \Pi_A B : U$:

```
ΠIcode-universal : ∀ {@♭ ℓ : Level} → ΠIData ℓ → U {ℓ}
ΠIcode-universal {ℓ} = code (ΠIData ℓ) (λ AB → (x : El (fst AB)) → El (snd AB x)) comΠ
```

Finally, precomposing into the universal code gives a more standard Π formation rule:

```
ΠIcode : {Γ : Set ℓ1} (A : Γ → U {ℓ2}) (B : Σ (El o A) → U {ℓ2}) → (Γ → U {ℓ2})
ΠIcode A B = ΠIcode-universal o (λ θ → (A θ , λ x → B (θ , x)))
```

3.2 Validating the axioms

Externally, the above internal language formalization implies the theorem mentioned in Section 1.3:

Theorem 2. *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}, which is closed under $=_{\mathbb{I}}$, $\vee$ and $\forall x : \mathbb{I}.-$. Then there is (for each size level i) a universe U_i classifying those semantic type families of size i equipped with a diagonal Kan composition structure (Definition 1) for generating cofibrations classified by Cof. U_i is closed under semantic Π, Σ, path, and glue types, and is itself Kan (U_{i+1} has a code for U_i). If $\hat{C}$ has cubical sets corresponding to boolean, natural number, and suspension types, then U_i is closed under those; if Cof is closed under $\wedge$, then U_i is closed under identity types as well.*

Proof. We interpret the logical framework in $\hat{C}$ in essentially the same way as in Orton and Pitts (2016): Agda's Π and Σ correspond to the dependent product and dependent sum of the presheaf topos $\hat{\mathbb{C}}$. In Agda we have written propositions using Π (for $\forall$), Σ (for $\wedge$), equality types (with postulated function extensionality and uniqueness of identity proofs), and a (postulated) "squashed" disjunction; we interpret these constructions using the subobject classifier of the presheaf topos, as outlined in Section 2.4. (Note that our Agda development is predicative, so we do not need the full power of the subobject classifier.) Finally, the Agda datatypes Nat and Bool are interpreted as the discrete cubical sets whose 0-cells are the natural numbers and booleans respectively.

It remains to validate the axioms used in the formalization. Relative to Orton and Pitts (2016), we have modified `ax5` (by making $r = r'$ a cofibration), omitted axioms `ax3`/`ax4` (which stipulate connections), and only use `ax7` (which states that cofibrations are closed under $\wedge$) in our construction of `Id` types with a judgemental computation rule on `refl`. 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 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 (2018) 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 Yoneda embedding of the interval $\mathbb{I}$ of C, and we assumed $(0/x) \neq (1/x)$ 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 a subset of the decidable sieves. So the result carries over because we assumed that cofibrations are decidable sieves, i.e. that *Cof* is a subobject of Ω_{dec}.
- Flat variables: Licata et al. (2018, Remark 4.1) argue that Agda's modal/flat/crisp type theory can be interpreted in any presheaf topos whose base category has a terminal object.
- Tininess (right adjoint to exponentiation by $\mathbb{I}$): in presheaves whose base category has finite products, all representables are tiny, and $\mathbb{I}$ is interpreted as $y_{\mathbb{I}}$.
- Cofibrations: we have assumed that *Cof* is closed under the cofibrations assumed in the formalization, $=_{\mathbb{I}}$, $\vee$ and $\forall x : \mathbb{I}.-$, and to construct identity types, $\wedge$.
- Finally, our postulated Susp type in Section 3.1.6 corresponds to an initial algebra for the introduction rules of Section 2.15. Coquand et al. (2018, Section 2.4) explicitly construct a number of very similar initial algebras, including a suspension type, for

the CCHM Kan operation, using a constructive version of the small object argument (Swan, 2016). □

The Cartesian cube category, $\mathbb{C}$, is freely generated by finite products and an interval object $\mathbb{I}$, with maps $0, 1 : 1 \to \mathbb{I}$ with $0 \neq 1$. So, to show that the above theorem applies to $\mathbb{C}$, all that remains is to check that the assumed object of cofibrations *Cof* can be constructed. Classically, one can take *Cof* to be Ω, using the usual logical operations of $=, \vee, \wedge, \forall$ for the subobject classifier of the topos. But because we work in a constructive metatheory, we must check that that there is an object of decidable sieves that is closed under these operations.

One way to do this, following Cohen et al. (2018) and Orton and Pitts (2016), is to use a "face lattice":

Proposition 1. *For the Cartesian cube category $\mathbb{C}$, $\hat{\mathbb{C}}$ contains an object Cof that is a subobject of Ω_{dec}, which is closed under $=_{\mathbb{I}}$, $\vee$, $\wedge$, and $\forall x : \mathbb{I}.-$.*

Proof. Recall (Orton and Pitts, 2016, Definition 6.2) that $\Omega_{dec}(\Psi)$ is the set of precomposition-closed collections of maps into Ψ with the property that for a given map $\rho : \Psi' \to_C \Psi$ it is decidable whether ρ is in the collection.

We define a "face lattice" to be the subobject of Ω closed under only $=_{\mathbb{I}}$ and $\vee$ and $\wedge$, and observe that $\forall x : \mathbb{I}.-$ is admissible for this fragment by a quantifier elimination argument, as in Cohen et al. (2018): define $\forall x : \mathbb{I}.x = x$ to be true, $\forall x : \mathbb{I}.x = r$ to be false if $x \neq r$, $\forall x : \mathbb{I}.r = r'$ to be $r = r'$ if $x \neq r, r'$, and $\forall x : \mathbb{I}.(\alpha \vee \beta) = (\forall x : \mathbb{I}.\alpha) \vee (\forall x : \mathbb{I}.\beta)$ and $\forall x : \mathbb{I}.(\alpha \wedge \beta) = (\forall x : \mathbb{I}.\alpha) \wedge (\forall x : \mathbb{I}.\beta)$.

To see that this face lattice is a subobject of Ω_{dec}, the main idea is that $r = r'$ is a decidable sieve, because $\hom_{\mathbb{C}}(\Psi, \mathbb{I})$ has decidable equality (as equality of maps is just syntactic identity of $\Psi \vdash r : \mathbb{I}$ terms), and $\wedge$ and $\vee$ of decidable sieves is always decidable. Orton and Pitts (2018) give an argument that (for the Cohen et al. (2018) notion of cofibration) this definition satisfies $\texttt{Cof} \hookrightarrow \Omega_{dec}$. □

At the other extreme, we can take `Cof` to be Ω_{dec} itself.

Proposition 2. *Suppose C is a finite product category with an object $\mathbb{I}$, and that for any object Ψ, the hom-set $\hom_C(\Psi, \mathbb{I})$ has decidable equality. Then in $\hat{C}$, Ω_{dec} is closed under $=_{\mathbb{I}}$, $\vee$, $\wedge$, and $\forall x : \mathbb{I}.-$.*

Proof. First, $=_{\mathbb{I}}$ lands in decidable sieves because it denotes equality of morphisms in $\hom_C(\Psi, \mathbb{I})$, and $\wedge$ and $\vee$ preserve decidability. Thus, it remains to check that Ω_{dec} is closed under $\forall x : \mathbb{I}.-$. An alternate characterization of Ω_{dec} is that it classifies monomorphisms $m : A \Rightarrow_{\hat{C}} B$ such that for all Ψ, $A(\Psi) \Rightarrow B(\Psi)$ has a decidable image, i.e., one can decide whether a given $b \in B(\Psi)$ 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., if $A \hookrightarrow B$ is a cofibration then $A^{\hom_{\mathbb{C}}(-,\mathbb{I})} \hookrightarrow B^{\hom_{\mathbb{C}}(-,\mathbb{I})}$ is a cofibration). Therefore, we must show that if $m : A \hookrightarrow B$ has decidable image for all Ψ, then so does $m^{\hom_C(-,\mathbb{I})} : A^{\hom_C(-,\mathbb{I})} \hookrightarrow B^{\hom_C(-,\mathbb{I})}$. But this is true because

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

is just a dimension shift. □

3.3 Interpreting the syntactic type theory

While we do not give a formal interpretation of the syntax from Section 2 in the model from Section 3, the intended interpretation in the internal language is as follows:

- Dimension contexts Ψ are interpreted as $\mathbb{I}^n$ where $n = |\Psi|$.
- Dimension terms $\Psi \vdash r : \mathbb{I}$ are interpreted as functions $[\![\Psi]\!] \to \mathbb{I}$.
- Dimension formulas $\Psi \vdash \phi\ \mathsf{formula}$ are interpreted as functions $[\![\Psi]\!] \to \mathsf{Set}$.
- Cofibrations $\Psi \vdash \alpha\ \mathsf{cofib}$ are interpreted as functions $\alpha : [\![\Psi]\!] \to \mathsf{Set}$ such that for all $x : [\![\Psi]\!]$, $\mathsf{Cofib}(\alpha(x))$.
- $\phi \vdash_\Psi \alpha$ is interpreted as a function $\Pi x{:}[\![\Psi]\!].[\![\phi]\!](x) \to [\![\alpha]\!](x)$.
- $\Psi; \phi \vdash \Gamma\ \mathsf{ctx}$ are interpreted as functions $(\Sigma x{:}[\![\Psi]\!].[\![\alpha]\!](x)) \to \mathsf{Set}_i$.
- $\Psi; \phi; \Gamma \vdash A\ \mathsf{Type}_i$ are interpreted as functions $(\Sigma y{:}(\Sigma x{:}[\![\Psi]\!].[\![\alpha]\!](x)).[\![\Gamma]\!](y)) \to \mathsf{U}_i$.
- $\Psi; \phi; \Gamma \vdash a : A$ are interpreted as functions $\Pi z{:}(\Sigma y{:}(\Sigma x{:}[\![\Psi]\!].[\![\alpha]\!](x)).[\![\Gamma]\!](y)).\mathsf{El}(A(z))$.
- Judgemental equality is interpreted by equality in the internal language (which we represent in Agda using propositional equality).

Every judgement is interpreted as flat/crisp/closed terms of the above types. The internal language's formation, introduction, elimination, and $\beta\eta$ rules for the interpretations of the types then match the syntactic formation, introduction, elimination, and $\beta\eta$ rules; the equations for the Kan operations correspond to the definitions used in the proof of Theorem 1.

3.4 Connection to the De Morgan Kan operation

The assumptions of the argument in Section 3.2 continue to hold if we add connections to our cube category (subject to the laws of a distributive lattice), and in the De Morgan cube category (which further adds reversal and De Morgan laws) of Cohen et al. (2018); in both cases, the cube category has finite products and an interval with appropriate structure. We can therefore interpret our construction in presheaves over either of these categories, and compare our Kan operation to the CCHM Kan operation in the same setting.

Consider De Morgan cubical sets, and take $r =_{\mathbb{I}} r'$, $\vee$, $\wedge$, and $\forall x : \mathbb{I}.-$ to be cofibrations for both Kan operations, so that we have the structure needed to carry out both our model and that of Cohen et al. (2018). In the terminology of this paper the CCHM Kan operation is $\mathtt{com}_A^{z:0\to1}\ (\alpha \mapsto t)\ (b)$; call a type *endpoint-Kan* if it is equipped with such an operation, and *diagonal-Kan* if it is equipped with our composition operation (Definition 1).

If a type is diagonal-Kan, then it is immediately endpoint-Kan as a special case. Conversely, using connections and reversal, 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 we obtain diagonal Kan composition $\mathtt{com}_A^{z:r\to r'}\ (\alpha \mapsto t)\ (b)$ for endpoint-Kan types by:

$$\mathtt{com}^{z:0\to1}_{A\langle \mathit{if}(z=0,r,r')/z\rangle}\ (\alpha \mapsto z.t\langle \mathit{if}(z = 0, r, r')/z\rangle, (r = r') \mapsto \_.b)\ (b)$$

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

Because any two composites of the same filling problem are homotopic, 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.

4. Recent and future work

Cubical type theories and cubical models of homotopy type theory have continued to receive a great deal of attention from the community since the original draft of this article in December 2017.

Regarding their syntactic metatheory, Huber (2018) and Angiuli (2019) proved canonicity for the De Morgan and Cartesian cubical type theories (respectively), while Sterling and Angiuli (2021) have recently proved the more challenging normalization property for Cartesian cubical type theory, an open-term analogue of canonicity that justifies the sophisticated type checking algorithms used by nearly all proof assistants for cubical type theory.

In extensions of the work in this article, Cavallo et al. (2020) have devised a common generalization of the diagonal Kan and De Morgan Kan operations in which the $r = r'$ equation is weakened to a path, extending the ideas in Section 3.4 to provide a more detailed comparison between the two. Weaver and Licata (2020) integrate ideas from our model and the De Morgan model to give a directed bicubical type theory in the style of Riehl and Shulman (2017), in which this paper's cubical type theory takes the place of homotopy type theory for the "space-like" indexing direction.

Work on Cartesian cubical type theory continues as well. Higher inductive types in cubical type theory have been studied in more generality, in the Cartesian setting by Cavallo and Harper (2019), who define a schema for higher inductive types, and also in the De Morgan setting by Coquand et al. (2018). Sterling et al. (2019) developed XTT, a Cartesian cubical type theory satisfying judgemental uniqueness of identity proofs, intended to give access to good extensionality principles in the absence of equality reflection; their work uses a variation of diagonal Kan composition that satisfies regularity (but not univalence).

Recently, the `cooltt` prototype proof assistant for Cartesian cubical type theory (`https://github.com/RedPRL/cooltt`) has added support for Π types indexed over $\mathbb{I}$ and `Cof` (with judgemental propositional univalence), in essence exposing more of the Orton–Pitts language of Section 3 in the syntax. Coupled with extension types in the style of Riehl and Shulman (2017), the resulting theory can fully internalize the type of `com`.

At a recent Dagstuhl seminar (Bauer et al., 2019), a team including some of the authors used the Agda formalization described here to check a new lemma designed for optimizing the Kan operation for the universe. Although being Kan is a homotopy proposition, there are different ways to define composition up to judgemental equality, and these differ in efficiency of implementation. (In fact, Coquand et al. (2019*a*) establish homotopy canonicity for De Morgan cubical type theory without type-specific judgemental equalities for Kan composition, showing that these choices differ *only* up to judgemental equality.)

Another major area of future work is to compare the various cubical models with each other, and with traditional settings for homotopy theory, such as simplicial sets. Such comparisons can be formulated using Quillen equivalence of model structures, because each Kan composition operation gives rise to a type-theoretic model structure (Sattler, 2017). One interesting question is whether the Cartesian cubical type theory defined here coincides with the one defined in Awodey (2018*b*,a)—this might provide insight into whether the regularity condition on the Kan operation influences the model structure.

Another important question is whether the model structure defined by the type theory is Quillen equivalent to the Kan model structure on simplicial sets, which implies that all synthetic homotopy theory in the cubical type theory can be transferred to facts about a standard notion of spaces. Coquand and Sattler have shown that the Cartesian and De Morgan model structures obtained in this way are *not* equivalent to simplicial sets (Sattler, 2018). However, Awodey, Cavallo, Coquand, Riehl, Sattler have recently shown that Cartesian cubical sets can be equipped with an "equivariant" Kan operation inducing a model structure that *is* Quillen equivalent to simplicial sets (Riehl, 2019). This work, which draws on the work of Awodey (2018*b*,a) and ourselves, extends composition from the $z : r \to r'$ scenario considered here to simultaneous compositions $(z_1, \ldots, z_n) : (r_1, \ldots, r_n) \to (r'_1, \ldots, r'_n)$ modulo permutation. The type theory in Section 2 admits a semantics in this equivariant Cartesian setting, viewing our `com` rule as a necessary but not sufficient condition of semantic fibrancy.

A different way to show that a cubical type theory is a suitable setting for synthetic homotopy theory would be to seek an interpretation in all ∞-toposes, which would not only imply an interpretation in simplicial sets, but justify a number of additional applications as well. Coquand et al. (2019*b*) make some progress on expanding the models of cubical type theory by giving a constructive definition of a class of sheaf models. Though they use the De Morgan model as the base category of spaces, it is also possible to use the Cartesian model for at least the special case of these models corresponding to the cobar construction (Weaver and Licata, 2020), and we plan to investigate the general case in future work.

References

Angiuli, C. 2019. *Computational Semantics of Cartesian Cubical Type Theory*. PhD thesis, Carnegie Mellon University. Technical Report CMU-CS-19-127. Available from http://reports-archive.adm.cs.cmu.edu/anon/2019/abstracts/19-127.html.

Angiuli, C., **Cavallo, E.**, **Hou (Favonia), K.-B.**, **Harper, R.**, **and Sterling, J.** 2018a. The RedPRL proof assistant (invited paper). In **Blanqui, F. and Reis, G.**, editors, *13th International Workshop on Logical Frameworks and Meta-Languages: Theory and Practice (LFMTP 2018)*, volume 274 of *Electronic Proceedings in Theoretical Computer Science*.

Angiuli, C. and Harper, R. 2017. Computational higher type theory II: dependent cubical realizability. Preprint. https://arxiv.org/abs/1606.09638.

Angiuli, C., **Harper, R.**, **and Wilson, T.** 2016. Computational higher type theory I: abstract cubical realizability. Preprint. http://arxiv.org/abs/1604.08873.

Angiuli, C., **Harper, R.**, **and Wilson, T.** 2017a. Computational higher-dimensional type theory. In *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages*, POPL 2017, pp. 680–693, New York, NY, USA. ACM.

Angiuli, C., **Hou (Favonia), K.-B.**, **and Harper, R.** 2017b. Computational higher type theory III: univalent universes and exact equality. Preprint. https://arxiv.org/abs/1712.01800.

Angiuli, C., **Hou (Favonia), K.-B.**, **and Harper, R.** 2018b. Cartesian cubical computational type theory: Constructive reasoning with paths and equalities. In *Computer Science Logic*.

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

Awodey, S. 2018a. An algebraic fibrant replacement for cubical sets. Talk at Category Theory OctoberFest.

Awodey, S. 2018b. A cubical model of homotopy type theory. *Annals of Pure and Applied Logic*, 169(12):1270–1294. Logic Colloquium 2015.

Bauer, A., **Escardó, M.**, **Lumsdaine, P. L.**, **and Mahboubi, A.** 2019. Formalization of Mathematics in Type Theory (Dagstuhl Seminar 18341). *Dagstuhl Reports*, 8(8):130–155.

Bentzen, B. 2019. Naive cubical type theory. Preprint. https://arxiv.org/abs/1911.05844.

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

Bezem, M. and Coquand, T. 2015. A Kripke model for simplicial sets. *Theoretical Computer Science*, 574:86–91.

Bezem, M., **Coquand, T.**, **and Huber, S.** 2014. A model of type theory in cubical sets. In **Matthes, R. and Schubert, A.**, editors, *19th International Conference on Types for Proofs and Programs (TYPES 2013)*, pp. 107–128.

Bezem, M., **Coquand, T.**, **and Huber, S.** 2018. The univalence axiom in cubical sets. *Journal of Automated Reasoning*.

Bickford, M. 2020. Formalization of cubical type theory in Nuprl. http://nuprl.org/KB/show.php?ID=774.

Birkedal, L., **Bizjak, A.**, **Clouston, R.**, **Grathwohl, H. B.**, **Spitters, B.**, **and Vezzosi, A.** 2018. Guarded cubical type theory. *Journal of Automated Reasoning*.

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

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

Cavallo, E. 2021. *Higher Inductive Types and Internal Parametricity for Cubical Type Theory*. PhD thesis, Carnegie Mellon University. Technical Report CMU-CS-21-100. Available from http://reports-archive.adm.cs.cmu.edu/anon/2021/CMU-CS-21-100.pdf.

Cavallo, E. and Harper, R. 2019. Higher inductive types in cubical computational type theory. *Proceedings of the ACM on Programming Languages*, 3(POPL):1:1–1:27.

Cavallo, E., **Mörtberg, A.**, **and Swan, A. W.** 2020. Unifying Cubical Models of Univalent Type Theory. In **Fernández, M. and Muscholl, A.**, editors, *28th EACSL Annual Conference on Computer Science Logic (CSL 2020)*, volume 152 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 14:1–14:17, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

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

Cohen, C., **Coquand, T.**, **Huber, S.**, **and Mörtberg, A.** 2018. Cubical type theory: A constructive interpretation of the univalence axiom. In **Uustalu, T.**, editor, *21st International Conference on Types for Proofs and Programs (TYPES 2015)*, pp. 5:1–5:34.

Constable, R. L., **Allen, S. F.**, **Bromley, H. M.**, **Cleaveland, W. R.**, **Cremer, J. F.**, **Harper, R. W.**, **Howe, D. J.**, **Knoblock, T. B.**, **Mendler, N. P.**, **Panangaden, P.**, **Sasaki, J. T.**, **and Smith, S. F.** 1986. *Implementing Mathematics with the NuPRL Proof Development System*. Prentice Hall.

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

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

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

Coquand, T., **Huber, S.**, **and Mörtberg, A.** 2018. On higher inductive types in cubical type theory. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS 2018, pp. 255–264, New York, NY, USA. ACM.

Coquand, T., **Huber, S.**, **and Sattler, C.** 2019a. Homotopy canonicity for cubical type theory. In **Geuvers, H.**, editor, *4th International Conference on Formal Structures for Computation and Deduction (FSCD 2019)*, volume 131 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 11:1–11:23, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Coquand, T., **Ruch, F.**, **and Sattler, C.** 2019b. Constructive sheaf models of type theory. Preprint. https://arxiv.org/abs/1912.10407.

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

Harper, R. 2018. Computational type theory. Lectures at Oregon Programming Languages Summer School 2018.

Harper, R. and Angiuli, C. 2018. Computational (higher) type theory. Tutorial at POPL 2018.

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

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

Huber, S. 2018. Canonicity for cubical type theory. *Journal of Automated Reasoning*.

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

Kan, D. M. 1955. Abstract homotopy. I. *Proceedings of the National Academy of Sciences of the United*

States of America, 41(12):1092–1096.

Kapulkin, K. and Lumsdaine, P. L. 2021. The simplicial model of Univalent Foundations (after Voevodsky). *Journal of the European Mathematical Society*, 23:2071–2126.

Licata, D. 2016. Weak univalence with "beta" implies full univalence. Email to Homotopy Type Theory mailing list. https://groups.google.com/d/msg/homotopytypetheory/j2KBIvDw53s/YTDK4D0NFQAJ.

Licata, D. R., **Orton, I.**, **Pitts, A. M.**, **and Spitters, B.** 2018. Internal universes in models of homotopy type theory. In *Formal Structures for Computation and Deduction (FSCD)*. arXiv:1801.07664.

Orton, I. and Pitts, A. M. 2016. Axioms for modelling cubical type theory in a topos. In **Talbot, J.-M. and Regnier, L.**, editors, *25th EACSL Annual Conference on Computer Science Logic (CSL 2016)*, volume 62 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 24:1–24:19, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Orton, I. and Pitts, A. M. 2018. Axioms for modelling cubical type theory in a topos (extended version). *Logical Methods in Computer Science*, 14(4).

Orton, R. I. 2019a. Agda code accompanying PhD thesis: Cubical Models of Homotopy Type Theory – An Internal Approach. https://doi.org/10.17863/CAM.35681.

Orton, R. I. 2019b. *Cubical Models of Homotopy Type Theory - An Internal Approach*. PhD thesis, University of Cambridge. https://doi.org/10.17863/CAM.36690.

Pitts, A. M. 2015. 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.

Riehl, E. 2019. The equivariant uniform Kan fibration model of cubical homotopy type theory. Talk at the International Conference on Homotopy Type Theory (HoTT 2019).

Riehl, E. and Shulman, M. 2017. A type theory for synthetic ∞-categories. *Higher Structures*, 1(1):147–224.

Sattler, C. 2017. The equivalence extension property and model structures. Preprint. https://arxiv.org/abs/1704.06911.

Sattler, C. 2018. Do cubical models of type theory also model homotopy types? Talk at Workshop on Types, Homotopy Type theory, and Verification at Hausdorff Institute.

Shulman, M. 2019. All $(\infty, 1)$-toposes have strict univalent universes. Preprint. https://arxiv.org/abs/1904.07004.

Sterling, J. and Angiuli, C. 2021. Normalization for cubical type theory. In *36th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '21, pp. 1–15.

Sterling, J., **Angiuli, C.**, **and Gratzer, D.** 2019. Cubical syntax for reflection-free extensional equality. In **Geuvers, H.**, editor, *4th International Conference on Formal Structures for Computation and Deduction (FSCD 2019)*, volume 131 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 31:1–31:25, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Swan, A. 2016. An algebraic weak factorization system on 01-substitution sets: A constructive proof. *Journal of Logic & Analysis*, 8(1).

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

Vezzosi, A., **Mörtberg, A.**, **and Abel, A.** 2019. Cubical Agda: A dependently typed programming language with univalence and higher inductive types. *Proceedings of the ACM on Programming Languages*, 3(ICFP):87:1–87:29.

Voevodsky, V. 2006. A very short note on homotopy λ-calculus. Unpublished. http://www.math.ias.edu/vladimir/files/2006_09_Hlambda.pdf.

Weaver, M. Z. and Licata, D. R. 2020. A constructive model of directed univalence in bicubical sets. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '20, pp. 915–928, New York, NY, USA. Association for Computing Machinery.