On Higher Inductive Types in Cubical Type Theory

Thierry Coquand
University of Gothenburg
thierry.coquand@cse.gu.se

Simon Huber
University of Gothenburg
simon.huber@cse.gu.se

Anders Mörtberg
Carnegie Mellon University and
University of Gothenburg
mortberg@cmu.edu

Abstract

Cubical type theory provides a constructive justification to certain aspects of homotopy type theory such as Voevodsky's univalence axiom. This makes many extensionality principles, like function and propositional extensionality, directly provable in the theory. This paper describes a constructive semantics, expressed in a presheaf topos with suitable structure inspired by cubical sets, of some higher inductive types. It also extends cubical type theory by a syntax for the higher inductive types of spheres, torus, suspensions, truncations, and pushouts. All of these types are justified by the semantics and have judgmental computation rules for all constructors, including the higher dimensional ones, and the universes are closed under these type formers.

Keywords Cubical Type Theory, Higher Inductive Types, Homotopy Type Theory, Univalent Foundations

1 Introduction

Homotopy type theory [26] provides a new and promising approach to equality in type theory where types are thought of as abstract spaces and equality as paths in these spaces [5]. Iterated equality proofs then correspond to *homotopies* between paths. This intuition is motivated by homotopy theoretic models, in particular by the Kan simplicial set model [15] due to Voevodsky. This allows one to find new principles in type theory inspired by homotopy theory. Prime examples of this are Voevodsky's *univalence axiom* [27], which generalizes the principle of propositional extensionality to dependent type theory, and the stratification of types by the complexity of their equality (i.e., by their homotopy level or "h-level" [28]).

In the homotopical interpretation of type theory inductive types are represented as discrete spaces with only points in them. Higher inductive types are a natural generalization where types may also be generated by paths (potentially higher dimensional). This notion of types, combined with universes and the univalence axiom, is an important extension of dependent type theory, which allows for an elegant and original synthetic development of algebraic topology, using in a key way type-theoretic ideas (such as the encode-decode method [26]). Impressive examples of this development are, among others, the definition of the Hopf fibration, the Freudenthal suspension theorem and the Blakers-Massey theorem [6, 13]. However, and somewhat surprisingly, despite several efforts (e.g., [19]), the *consistency* of such an extension, which would justify these impressive developments, has not yet been established. The simplicial set model [15] provides (in a classical framework) a model for the univalence axiom, but it only provides a model for some very particular higher inductive types (such as the spheres, and the propositional truncation via an impredicative encoding [28]), and, as explained in [19], it is not clear how to extend this model to a model of parametrized higher inductive types like the suspension or pushouts (expressed as operations on a given universe).

Contributions The first contribution of the present paper is to provide such a semantics, starting in an essential way not from the simplicial set model, but from a cubical set model [8, 20]. This semantics is furthermore carried out in a constructive meta-theory. Our second contribution is to extend cubical type theory with a syntax for higher inductive types, exemplified by: spheres, the torus, suspensions, truncations, and pushouts. These types illustrate many of the difficulties in giving a computational justification for a general class of higher inductive types, in particular: the spheres and torus have higher dimensional constructors, furthermore one version of the torus refers to "fibrancy" structure in its endpoints, the suspension has a parameter type, the truncations are recursive, and the pushouts have function applications in the endpoints of the path constructor. We show how to overcome all of these difficulties in a uniform way which suggests an approach to the problem of defining a schema for higher inductive types in cubical type theory.

Furthermore, all of the higher inductive types we consider have the following good properties justified by our semantics:

1. judgmental computation rules for all constructors,
2. strict stability under substitution, and
3. closure under universe levels (the higher inductive types live in the same universe as their parameters).

We have also implemented a variation of the system presented in this paper and performed multiple experiments with it.[1]

Outline The paper begins by describing the semantics, expressed in a presheaf topos with suitable structure, of the circle (Section 2.1), suspension (Section 2.2), and pushouts (Section 2.3). The next section starts with a short background on cubical type theory (Section 3.1) followed by the extension to the theory with: circle and spheres (Section 3.3.1), the torus (Section 3.3.2), suspensions (Section 3.3.3), propositional truncation (Section 3.3.4), and pushouts (Section 3.3.5). The paper ends with conclusions and discussions on future and related work (Section 4).

2 Semantics of higher inductive types

As shown in [2, 18, 20], the presentation of the semantics of cubical type theory can be both simplified and clarified by using the language of extensional type theory (with universes). This language can be given meaning in any presheaf topos, so long as we assume

LICS '18, July 9–12, 2018, Oxford, United Kingdom

ACM ISBN 978-1-4503-5583-4/18/07…$15.00
https://doi.org/10.1145/3209108.3209197

[1] See: https://github.com/mortberg/cubicaltt/tree/hcomptrans

that the ambient set theory has a hierarchy of Grothendieck universes. In particular, we are going to show that the justification of higher inductive types can be done internally, using the existence of suitable initial algebras as the only extra assumption. We then justify the existence of these initial algebras for our presheaf topos externally. The key idea will be a decomposition of the notion of composition structure [18, 20] into a *transport* and a *homogeneous composition* operation.[2] This decomposition can be described internally.

We will work here in the presheaf topos over the Lawvere theory of De Morgan algebras [8, 18] (but, following [20], our results are valid in a more general setting). The presentation we use in [8] of this category is the following: we fix a countable set of names/symbols and the objects of the category $I, J, \ldots$ are finite sets of symbols. A map $J \to I$ is then a set-theoretic map from I to the free De Morgan algebra $\mathsf{dM}(J)$ on J. The corresponding presheaf model has then a generic De Morgan algebra $\mathbb{I}$, taking $\mathbb{I}(J)$ to be $\mathsf{dM}(J)$. (To have such a structure on $\mathbb{I}$ is not strictly necessary [20], but it simplifies the presentation.)

This type $\mathbb{I}$ is used as an abstract representation of the unit interval, so that a path in a type A is represented by an element of the exponential $A^{\mathbb{I}}$. The extra data needed to define a cubical set model is a notion of *cofibration*, which specifies the shape of filling problems that can be solved in a dependent type. We represent this by a type of *cofibrant* propositions $\mathbb{F}$ (denoted by `Cof` in [20]). In [8], this is represented by the *face lattice* (see Section 3.1), but other choices are possible. (Classically, this type $\mathbb{F}$ is a subtype of the subobject classifier of the presheaf topos, but, as stressed in [18], we can avoid mentioning the impredicative type of propositions altogether, and work in a predicative meta-theory.) We write $[\varphi]$ for the type associated to the proposition $\varphi : \mathbb{F}$. So $[\varphi]$ is a sub-singleton, and any element of $[\varphi]$ is equal to a fixed element $\mathtt{tt}$.

A *partial element* of a type T is given by an element φ in $\mathbb{F}$ and a function $[\varphi] \to T$. We say that a total element v of T extends such a partial element φ, u if we have $\varphi \Rightarrow u\,\mathtt{tt} = v$, where $\Rightarrow$ denotes implication between propositions.

In this extensional type theory, we can think of a dependent type A over a given type Γ as a family of types $A\rho$ indexed by elements ρ of Γ.

We now recall the notions of composition and filling structures [8, 20]. Let A be a dependent type over a type Γ.

Definition 2.1. A *composition structure* c_A on A is an operation taking as inputs γ in $\Gamma^{\mathbb{I}}$, a proposition φ in $\mathbb{F}$, a partial element u in $[\varphi] \to \Pi(i : \mathbb{I})\, A\gamma(i)$, and an element u_0 in $A\gamma(0)$ such that $\varphi \Rightarrow u\,\mathtt{tt}\,0 = u_0$. This operation produces an element $u_1 = \mathsf{c}_A\ \gamma\,\varphi\,u\,u_0$ in $A\gamma(1)$ such that $\varphi \Rightarrow u\,\mathtt{tt}\,1 = u_1$.

The type of all such operations is written $\mathsf{Comp}(\Gamma, A)$ (see [20, Definition 4.3] for an explicit internal definition).

Definition 2.2. A *filling structure* f_A on A is an operation taking the same input as c_A above, but producing an element $v = \mathsf{f}_A\ \gamma\,\varphi\,u\,u_0$ in $\Pi(i : \mathbb{I})\, A\gamma(i)$ such that v extends u, i.e., $\varphi \Rightarrow u\,\mathtt{tt} = v$, and $v\ 0 = u_0$.

We write $\mathsf{Fill}(\Gamma, A)$ for the type of filling structures on A.

[2] As explained in [2] this decomposition was first introduced in an early version of [8], precisely to address the problem of the semantics of propositional truncation and this decomposition is also present in [3, 4, 7].

This notion of filling structure is an internal form of the homotopy extension property, which was recognized very early (see, e.g., [11]) as a key for an abstract development of algebraic topology.

As explained in [8, 20] we have that $\mathsf{Comp}(\Gamma, A)$ is a retract of $\mathsf{Fill}(\Gamma, A)$.

In the particular case where Γ is the unit type, then A is a "global" type, and $\mathsf{Comp}(\Gamma, A)$ becomes the type $\mathsf{Fibrant}(A)$ expressing that A is a *fibrant object*. Such a *fibrancy structure* on A consists of an operation h_A taking as arguments u_0 in A and a partial element φ, u of $A^{\mathbb{I}}$ such that $\varphi \Rightarrow u\ \mathtt{tt}\ 0 = u_0$, and produces an element $u_1 = h_A\ \varphi\ u\ u_0$ such that $\varphi \Rightarrow u\ \mathtt{tt}\ 1 = u_1$.

In general, if A is a family of types over Γ, to give a composition structure for each fiber, that is, an element in $\Pi(\rho : \Gamma)\ \mathsf{Fibrant}(A\rho)$, is not enough to get a global composition structure, that is, an element in $\mathsf{Comp}(\Gamma, A)$ (see [20] for an explicit counterexample). An element in $\Pi(\rho : \Gamma)\ \mathsf{Fibrant}(A\rho)$ is called a *homogeneous composition structure*.

We now describe the notion of *transport* operation, which allows to define a composition structure from a homogeneous composition structure. This decomposition of the composition operation into a transport and homogeneous composition operation plays a crucial role for interpreting higher inductive types depending on parameters (like suspension, pushouts, or propositional truncation).

Definition 2.3. A *transport structure* t_A on A is an operation taking as arguments a path γ in $\Gamma^{\mathbb{I}}$, a proposition φ in $\mathbb{F}$ such that $\varphi \Rightarrow \forall (i : \mathbb{I})\, \gamma(0) = \gamma(i)$, and an element u_0 in $A\gamma(0)$. This operation produces an element $u_1 = t_A\,\gamma\,\varphi\,u_0$ in $A\gamma(1)$ such that $\varphi \Rightarrow u_0 = u_1$.

The condition $\varphi \Rightarrow \forall (i : \mathbb{I})\, \gamma(0) = \gamma(i)$ expresses that the path γ is *constant on* φ.

Clearly we obtain a homogeneous composition structure from any composition structure. We also get:

Lemma 2.4. *If a family of types A over Γ has a composition structure c_A, then it has a transport structure t_A.*

Proof. We can take $t_A\,\gamma\,\varphi\,u_0 = \mathsf{c}_A\ \gamma\,\varphi\,(\lambda(x : [\varphi])(i : \mathbb{I})\,u_0)\,u_0$. □

Lemma 2.5. *If a family of types A over Γ has a homogeneous composition structure h_A and a transport structure t_A, then it has a composition structure c_A.*

Proof. We can define $\mathsf{c}_A\ \gamma\,\varphi\,u\,u_0$ as

$$h_A\,\gamma(1)\,\varphi\,(\lambda(x : [\varphi])(i : \mathbb{I}).\ t_A\,\gamma'(i)\,(i = 1)\,(u\,x\,i))\,(t_A\,\gamma\,0_{\mathbb{F}}\,u_0)$$

where $\gamma'(i) = \lambda(j : \mathbb{I})\,\gamma(i \vee j)$. □

We are now going to develop some universal algebra internally in the presheaf model. The operations will involve the interval object $\mathbb{I}$ and the type $\mathbb{F}$ of cofibrant propositions, and can be seen as a generalization of the usual notion of operations in universal algebra.

2.1 Semantics of the circle

The circle is represented as a higher inductive type with a path loop connecting a point base to itself.

If A (resp. B) has a fibrancy structure h_A (resp. h_B), then a map $\alpha : A \to B$ is *fibrancy preserving* if it satisfies

$$\alpha\ (h_A\ \varphi\ u\ u_0) = h_B\ \varphi\ (\lambda(x : [\varphi])(i : \mathbb{I})\ \alpha\ (u\ x\ i))\ (\alpha\ u_0).$$

An S^1*-algebra structure* on a type A consists of a fibrancy structure h_A together with a base point b_A and a loop l_A in $A^{\mathbb{I}}$ connecting b_A to itself (i.e., $l_A\ 0 = l_A\ 1 = b_A$). Given two S^1-algebras A, h_A, b_A, l_A and B, h_B, b_B, l_B a function $\alpha : A \to B$ is a map of S^1-algebras if it is fibrancy preserving and satisfies $\alpha\ b_A = b_B$ and $\alpha\ (l_A\ i) = l_B\ i$.

We will show below using *external* reasoning:

Proposition 2.6. *There exists an initial S^1-algebra, which will be denoted by* $\mathbb{S}^1$, hcomp, base, loop.

So $\mathbb{S}^1$ has a structure of an S^1-algebra and the fact that it is initial means that, for any S^1-algebra A, h_A, b_A, l_A there exists a unique S^1-algebra map $\mathbb{S}^1 \to A$.

By definition, the type $\mathbb{S}^1$ is fibrant since it has a fibrancy structure hcomp. Furthermore, we can prove that initiality implies the dependent elimination rule.[3]

Proposition 2.7. $\mathbb{S}^1$ *satisfies the dependent elimination rule for the circle: given a family of types P over $\mathbb{S}^1$ with a composition structure, and a in P* base *and $l\ i$ in P* (loop i) *such that $l\ 0 = l\ 1 = a$ there exists a map* elim $: \Pi(x : \mathbb{S}^1)P\ x$ *such that* elim base $= a$ *and* elim (loop i) $= l\ i$.

Proof. We know by [8, 20] that $A = \Sigma(x : \mathbb{S}^1)P\ x$ has a composition structure. It has then a natural $\mathbb{S}^1$-algebra structure, taking $b_A =$ base, a and $l_A\ i =$ loop $i, l\ i$. This structure is such that the first projection $\pi_1 : A \to \mathbb{S}^1$ is a map of S^1-algebras. We have a unique S^1-algebra map $\alpha : \mathbb{S}^1 \to A$ and $\pi_1 \circ \alpha$ is the identity on $\mathbb{S}^1$. We can then define elim $x = \pi_2\ (\alpha\ x)$ in $P\ x$. □

2.2 Semantics of the suspension operation

The suspension Susp A of a type A has constructors N and S (two poles) and a path between them for any element of A. This enables us to give a direct definition of $\mathbb{S}^{n+1}$ as $\mathsf{Susp}^n\mathbb{S}^1$. Compared to the circle, this higher inductive type presents the extra complexity of having parameters and the decomposition of the composition operation will be the key for providing its semantics.

Given a type X, a Susp X*-algebra structure* on a type A consists of a fibrancy structure h_A together with two points n_A, s_A, and a family of paths l_A in $X \to A^{\mathbb{I}}$ connecting n_A to s_A (i.e., $l_A\ x\ 0 = n_A$ and $l_A\ x\ 1 = s_A$ for all x in X). Given two Susp X-algebras A, h_A, n_A, s_A, l_A and B, h_B, n_B, s_B, l_B a function $\alpha \colon A \to B$ is a map of Susp X-algebras if it is fibrancy preserving and satisfies $\alpha\ n_A = n_B$, $\alpha\ s_A = s_B$, and $\alpha\ (l_A\ i) = l_B\ i$.

As for the circle we can show using external reasoning:

Proposition 2.8. *There exists an initial* Susp X*-algebra, which will be denoted by* Susp X, hcomp, N, S, merid.

By definition, the type Susp X is fibrant since it has a fibrancy structure hcomp. Using this filling structure, we prove as above:

Proposition 2.9. Susp X *satisfies the dependent elimination rule for the suspension: given a family of type P over* Susp X *with a composition structure, and n in P* N *and s in P* S *and $l\ x\ i$ in P* (merid $x\ i$) *such that $l\ x\ 0 = n$ and $l\ x\ 1 = s$ there exists a map* elim $: \Pi(x : \mathsf{Susp}\,X)P\ x$ *such that* elim N $= n$ *and* elim S $= s$ *and* elim (merid $x\ i$) $= l\ x\ i$.

[3] This is a direct generalization of the usual argument that a natural number object satisfies the dependent elimination rule.

The operation Susp X is functorial in X. Given a map $u : X \to Y$ we get a Susp X-structure on Susp Y by taking $l_{\mathsf{Susp}\,Y}\ x\ i =$ merid$_Y$ $(u\ x)\ i$ and hence a map Susp(u) : Susp $X \to$ Susp Y.

Let now A be a dependent family of types over a given type Γ, so that $A\rho$ is a type for any ρ in Γ. We define a new family of types Susp A over Γ by taking $(\mathsf{Susp}\,A)\rho = \mathsf{Susp}(A\rho)$. By construction, this new family *always* has a homogeneous composition structure (without any hypothesis on A).

Proposition 2.10. *If A has a transport structure t_A, then* Susp A *has a transport structure, and hence (since it has a homogeneous composition structure) also a composition structure by Lemma 2.5.*

Proof. Given γ in $\Gamma^{\mathbb{I}}$ and φ such that γ is constant on φ (i.e., $\varphi \Rightarrow \forall(i : \mathbb{I})\ \gamma(0) = \gamma(i)$), we have a map $t_A\ \gamma\ \varphi : A\gamma(0) \to A\gamma(1)$ which is the identity on φ and hence the map Susp $(t_A\ \gamma\ \varphi)$ is a transport map Susp $(A\gamma(0)) \to$ Susp $(A\gamma(1))$ which is the identity on φ. □

This example motivates the decomposition of the composition operation into a transport and homogeneous composition operations. In a context, we could only build an initial algebra for the *homogeneous* composition operation (by doing it pointwise) and it does not seem possible to do it for the composition operation directly. The problem does not appear for a type like the circle which has no parameters, for which homogeneous and general compositions coincide. For the suspension however, we have to argue further that we also get a transport operation. (This problem seems connected to the problem of size blow-up for parametrized higher inductive types due to fibrant replacement in the simplicial set model discussed in [19].)

The same argument applies to the propositional truncation $\|X\|$ of a type X. We would then instead consider the following notion of algebra: a type A with a fibrancy structure, a map $i_A : X \to A$ and a map $sq_A : A \to A \to A^{\mathbb{I}}$ such that $sq\ a_0\ a_1$ is a path connecting a_0 to a_1.

2.3 Pushouts

Many examples of higher inductive types can be encoded as (homotopy) pushouts of spans of other types. In particular (homotopy) coequalizers, which together with coproducts (which are encoded using Σ-types), can be used to compute general colimits of diagrams of types. This has been used to encode many known higher inductive types, including recursive ones like propositional [9, 16] and higher truncations [22].

The semantics of pushouts involves the same problem with parameters as in the previous example, but the definition of the transport function is more complex and we will need to introduce some auxiliary operations definable from transport.

A *span* $D = (C, A, B, u, v)$ consists of two maps $u \colon C \to A$ and $v \colon C \to B$. Given such a span, we define a D-algebra to be a type X with a fibrancy structure h_X and maps $i_X \colon A \to X$ and $j_X \colon B \to X$ and $p_X \colon C \to X^{\mathbb{I}}$ such that $p_X\ z\ 0 = i_X\ (u\ z)$ and $p_X\ z\ 1 = j_X\ (v\ z)$. As above, there is a canonical notion of D-algebra maps, and (in suitable presheaf models) we have an initial D-algebra, which we write $\mathsf{po}(D) = A \sqcup_C B$, hcomp, inl, inr, push.

We can relativize this situation over a type Γ. If A, B, C are families of types over Γ and u (resp. v) is a family of maps $u\rho \colon C\rho \to A\rho$ (resp. $v\rho \colon C\rho \to B\rho$), we consider $D = (C, A, B, u, v)$ to be a span over Γ, with $D\rho = (C\rho, A\rho, B\rho, u\rho, v\rho)$. If the span D is given over

Γ, we define $\mathrm{po}(D)$ in a pointwise way as for the suspensions, taking $\mathrm{po}(D)\rho$ to be $\mathrm{po}(D\rho)$.

We want to prove that if C, A, B have transport structures, then so does $\mathrm{po}(D)$. In order to do that, we first show how to define further operations from a given transport structure.

Lemma 2.11. *Given a family of types A over Γ with a transport structure t_A we can define a new operation f_A such that $f_A\ \varphi\ \gamma\ a_0$ is a path in $\Pi(i : \mathbb{I})A\gamma(i)$ constant on φ and connecting a_0 to $t_A\ \gamma\ \varphi\ a_0$ for any γ in $\Gamma^{\mathbb{I}}$ constant on φ and a_0 in $A\gamma(0)$. Furthermore given any a in $\Pi(i : \mathbb{I})A\gamma(i)$ we can define an operation $sq_A\ \varphi\ \gamma\ a$ which is a path in $(A\gamma(1))^{\mathbb{I}}$ connecting $t_A\ \gamma\ \varphi\ a(0)$ to $a(1)$, and which is constant on φ.*

Proof. We define

$$f_A\ \varphi\ \gamma\ a_0 = \lambda(i : \mathbb{I})\, t_A\ (\lambda(j : \mathbb{I})\gamma(i \wedge j))\ (\varphi \vee (i = 0))\ a_0$$

which connects a_0 to $t_A\ \gamma\ \varphi\ a_0$ and is constant on φ, and

$$sq_A\ \varphi\ \gamma\ a = \lambda(i : \mathbb{I})\, t_A\ (\lambda(j : \mathbb{I})\gamma(i \vee j))\ (\varphi \vee (i = 1))\ a(i)$$

which connects $t_A\ \gamma\ \varphi\ a(0)$ to $a(1)$ and is constant on φ. □

The relationship between these operations can be displayed as:

$$\begin{tikzcd}
 & a(1) \\
a(0) \arrow[ur, "a"] \arrow[r, "f_A\ \varphi\ \gamma\ a(0)"'] & t_A\ \gamma\ \varphi\ a(0) \arrow[u, "sq_A\ \varphi\ \gamma\ a"'] \\
\gamma(0) \arrow[r, "\gamma"'] & \gamma(1)
\end{tikzcd}$$

so that sq_A can be though of as an operation which "squeezes" the path a into the fiber over $\gamma(1)$.

Corollary 2.12. *Given two families of types C and A over Γ with transport structures t_C and t_A respectively, and a map $u : C \to A$ over Γ, there exists an operation $l\ \varphi\ \gamma\ c_0$ which is a path in $(A\gamma(1))^{\mathbb{I}}$ constant over φ and connecting $t_A\ \gamma\ \varphi\ (u\gamma(0)\ c_0)$ and $u\gamma(1)\ (t_C\ \gamma\ \varphi\ c_0)$, given γ in $\Gamma^{\mathbb{I}}$ constant over φ and c_0 in $C\gamma(0)$.*

Proof. We apply the sq_A operation and the f_C operation from Lemma 2.11 to the path $\lambda(i : \mathbb{I})\, u\gamma(i)\, (f_C\ \varphi\ \gamma\ c_0\ i)$. □

Proposition 2.13. *Given a family of spans $D = (C, A, B, u, v)$ over a type Γ such that A, B, and C have transport structures then the family $\mathrm{po}(D)$ also has a transport structure, and hence also a composition structure by Lemma 2.5.*

Proof. We use the previous corollary to provide a structure of $D\gamma(0)$-algebra on $\mathrm{po}(D)\gamma(1)$, structure which coincides with the one of $\mathrm{po}(D)\gamma(0)$ on φ. By initiality we get a map $\mathrm{po}(D)\gamma(0) \to \mathrm{po}(D)\gamma(1)$ which is the identity on φ, and is the desired transport function. (For a more detailed explanation see the syntactic presentation in Section 3.3.5.) □

2.4 Existence of initial algebras

We now explain the proof of Proposition 2.6 asserting the existence of a suitable initial algebra. We cannot prove this in an abstract way, but we need to use the fact that we are working with presheaf models over a small base category $\mathcal{C}$, in our case the Lawvere theory of the theory of De Morgan algebras. We write $I, J, K, \ldots$ for the objects of $\mathcal{C}$. We only describe the case of S^1-algebra here, but all other cases follow the same pattern.[4] The argument we give can be seen as a constructive version of the small object argument [25], and it crucially uses the fact that both $\mathbb{F}(I)$ and $\mathbb{I}(I)$ have decidable equality. Classically we could use Garner's small object argument [12] as is for instance done in [19].

We first define inductively a family of sets $\mathbb{S}^1_{\mathrm{pre}}(I)$ which is an "upper approximation" of the circle, together with maps $\mathbb{S}^1_{\mathrm{pre}}(I) \to \mathbb{S}^1_{\mathrm{pre}}(J)$, $u \mapsto uf$ for $f : J \to I$. An element of $\mathbb{S}^1_{\mathrm{pre}}(I)$ is of the form:

- base, or
- $\mathsf{loop}\ r$ with $r \neq 0, 1$ in $\mathbb{I}(I)$, or
- $\mathsf{hcomp}\ [\varphi \mapsto u]\ u_0$ with $\varphi \neq 1$ in $\mathbb{F}(I)$ and u_0 in $\mathbb{S}^1_{\mathrm{pre}}(I)$ and u a family of elements $u_{f,r}$ in $\mathbb{S}^1_{\mathrm{pre}}(J)$ for $f : J \to I$ such that $\varphi f = 1$ and r in $\mathbb{I}(J)$.

In this way an element of $\mathbb{S}^1_{\mathrm{pre}}(I)$ can be seen as a well-founded tree. Note that we do not yet require that the sides in hcomp match up with the base. In order to express this we first define uf in $\mathbb{S}^1_{\mathrm{pre}}(J)$ for $f : J \to I$ by induction on u:

$$\mathsf{base} f = \mathsf{base}$$

$$(\mathsf{loop}\ r)f = \begin{cases} \mathsf{loop}\ (rf) & \text{if } rf \neq 0 \text{ and } rf \neq 1 \\ \mathsf{base} & \text{otherwise} \end{cases}$$

$$(\mathsf{hcomp}\ [\varphi \mapsto u]\ u_0)f = \begin{cases} u_{f,1} & \text{if } \varphi f = 1 \\ \mathsf{hcomp}\ [\varphi f \mapsto uf^+]\ (u_0 f) & \text{otherwise} \end{cases}$$

where uf^+ is the family $(uf^+)_{g,r} = u_{fg,r}$ for $g\colon K \to J$.

Note that we may not have in general $(vf)g = v(fg)$ for v in $\mathbb{S}^1_{\mathrm{pre}}(I)$ and $f : J \to I$ and $g : K \to J$. We then inductively define the subsets $\mathbb{S}^1(I) \subseteq \mathbb{S}^1_{\mathrm{pre}}(I)$ by taking the elements base, loop r, and $\mathsf{hcomp}\ [\varphi \mapsto u]\ u_0$ such that $u_0 \in \mathbb{S}^1(I)$, $u_{f,r} \in \mathbb{S}^1(J)$, for $f : J \to I$ satisfying $u_0 g = u_{g,0}$ for $g : J \to I$ and $u_{f,r} g = u_{fg,rg}$ for $f : J \to I$ and r in $\mathbb{I}(J)$ and $g : K \to J$. This defines a cubical set $\mathbb{S}^1$, such that $\mathbb{S}^1(I)$ is a subset of $\mathbb{S}^1_{\mathrm{pre}}(I)$ for each I.

As defined $\mathbb{S}^1$ has a structure of an S^1-algebra. Let us sketch that $\mathbb{S}^1$ is also the initial S^1-algebra in this presheaf model. Note that initiality stated internally is a statement quantifying over all possible types in a universe, which for simplicity we did not make explicit. Unfolding this internal quantification amounts to constructing (suitably unique) natural transformations $\mathsf{elim}\colon \mathbb{S}^1 \to A$ where A is a presheaf over the category of elements of $\mathbf{y}(I)$ equipped with a homogeneous composition structure and sections b in A and l in $A^{\mathbb{I}}$ connecting b to itself; moreover, these natural transformations elim should be stable under substitutions $\mathbf{y}(f)\colon \mathbf{y}(J) \to \mathbf{y}(I)$. This works more generally for A being a presheaf over any cubical set Γ, not only representables: $\mathsf{elim}\,\rho\, u$ in $X(I, \rho)$ for ρ in $\Gamma(I)$ and u in $\mathbb{S}^1(I)$ is defined by induction on the height of the well-founded tree u simultaneously with verifying $(\mathsf{elim}\,\rho\, u)f = \mathsf{elim}\,(\rho f)\,(uf)$ for $f : J \to I$. Note that the height of uf does not increase. Each case in the definition is guided by the uniqueness condition.

2.5 Universes

As shown externally in [8, 20] (and internally in [18]) we can define in the presheaf model a cumulative hierarchy of (univalent and fibrant) universes U_n which classify families of types of a given size with a composition structure. Since the way we build initial

[4]The interested reader may consult the Appendix of https://arxiv.org/abs/1802.01170 for the proofs for the other higher inductive types.

algebras preserves the universe level, our definition, e.g., of the suspension can be seen as an operation $\mathsf{Susp} : U_n \to U_n$.

Let us expand this point. Let $\mathcal{U}_n$ be a cumulative sequence of Grothendieck universes (or constructive analog of them [1]) in the underlying set theory. If Γ is a presheaf on C and A a $\mathcal{U}_n$-valued presheaf on the category of elements of Γ with a composition structure c_A, the suspension operation builds a $\mathcal{U}_n$-valued presheaf $\mathsf{Susp}\,A$ with composition structure $\mathsf{Susp}\,c_A$ such that if $\sigma : \Delta \to \Gamma$ we have $(\mathsf{Susp}\,A)\sigma = \mathsf{Susp}\,(A\sigma)$ and $(\mathsf{Susp}\,c_A)\sigma = \mathsf{Susp}\,(c_A\sigma)$. An element of $U_n(I)$ is then a pair A, c_A where A is a $\mathcal{U}_n$-valued presheaf on the category of elements of $\mathbf{y}(I)$ and c_A a composition structure on A, and Susp can then be seen as a natural transformation $U_n \to U_n$.

Thus, we have presented a semantics of a large class of higher inductive types with univalent universes. (As shown in [26], the univalence axiom is essential for any non trivial use of the higher-dimensional structure of higher inductive types.)

3 Higher inductive types in cubical type theory

In this section we discuss the extensions to cubical type theory by higher inductive types. We begin by recalling the basic notions of cubical type theory [8].

3.1 Background: cubical type theory

Cubical type theory extends a dependent type theory with a universe U closed under Π- and Σ-types with Path-types, composition operations and Glue-types.

The Path-types internalize the idea from homotopy type theory that equalities correspond to paths. We write $\mathsf{Path}\,A\,a\,b$ for the type of paths in A with endpoints a and b. These types behave like function types and have both abstraction (written $\langle i\rangle\, t$ for t with i abstracted) and application (written using juxtaposition). The path abstraction binds "dimension variables" ranging over an abstract interval $\mathbb{I}$ specified by the grammar:

$$r, s \quad ::= \quad 0 \mid 1 \mid i \mid 1 - r \mid r \wedge s \mid r \vee s$$

The set $\mathbb{I}$ is a De Morgan algebra with the $1 - r$ operation as De Morgan involution. A type in a context with dimension variables $i_1, \ldots, i_n : \mathbb{I}$ should be thought of as an n-dimensional cube and the substitutions $(i/0)$ and $(i/1)$ give the faces of this cube. A substitution (i/j) renames the dimension variable i in A into j and as there are no injectivity constraints on these renaming substitutions one can perform substitutions which give a "diagonal" of a cube (i.e., if A is a square depending on $i, j : \mathbb{I}$, then $A(i/j)$ is a diagonal). The $\wedge$ and $\vee$ operations are called *connections* and provide convenient ways of building higher dimensional cubes from lower dimensional ones. For instance, if A is a line depending on i, then $A(i/i \wedge j)$ is the interior of the square:

$$\begin{tikzcd}
A(i/0)(j/1) \arrow[r, "A(i/i)"] & A(i/1)(j/1) & & j \\
A(i/0)(j/0) \arrow[u, "A(i/0)"] \arrow[r, "A(i/0)"'] \arrow[ur, phantom, "A(i/i \wedge j)"] & A(i/1)(j/0) \arrow[u, "A(i/j)"'] & & \cdot \arrow[u] \arrow[r] & i
\end{tikzcd}$$

The face lattice $\mathbb{F}$ is a distributive lattice generated by formal symbols $(i = 0)$ and $(i = 1)$ with the relation $(i = 0) \wedge (i = 1) = 0_{\mathbb{F}}$. The elements of the face lattice can be described by the grammar:

$$\varphi, \psi \quad ::= \quad 0_{\mathbb{F}} \mid 1_{\mathbb{F}} \mid (i = 0) \mid (i = 1) \mid \varphi \wedge \psi \mid \varphi \vee \psi$$

There is a canonical lattice map $\mathbb{I} \to \mathbb{F}$ sending i to $(i = 1)$ and $1 - i$ to $(i = 0)$. We write $(r = 1)$ for the image of $r : \mathbb{I}$ in $\mathbb{F}$ and we write $(r = 0)$ for $((1 - r) = 1)$.

The judgment $\Gamma \vdash \varphi : \mathbb{F}$ says that φ is a face formula involving only the dimension variables declared in Γ. Given a formula φ we can *restrict* a context Γ and obtain a new context written Γ, φ (assuming that φ only depends on the dimension variables in Γ). We call terms and types in such a restricted context *partial.* These restricted contexts are used for specifying the boundary of higher dimensional cubes, for example, if A is a line depending on i, the partial type $i : \mathbb{I}, (i = 0) \vee (i = 1) \vdash A$ is the two endpoints of A. If $\Gamma, \varphi \vdash v : A$, we write $\Gamma \vdash u : A[\varphi \mapsto v]$ to denote the two judgments:

$$\Gamma \vdash u : A \qquad\qquad \Gamma, \varphi \vdash u = v : A$$

Using this we can express the typing rule for the composition operations:

$$\frac{\Gamma \vdash \varphi : \mathbb{F} \qquad \Gamma, \varphi, i : \mathbb{I} \vdash u : A \qquad \begin{array}{c}\Gamma, i : \mathbb{I} \vdash A \\ \Gamma \vdash u_0 : A(i/0)[\varphi \mapsto u(i/0)]\end{array}}{\Gamma \vdash \mathsf{comp}^i\, A\, [\varphi \mapsto u]\, u_0 : A(i/1)[\varphi \mapsto u(i/1)]}$$

This operation takes a line type A, a formula φ, a partial line term u and a term u_0 of type $A(i/0)$ (note that i may occur freely in A and u). Furthermore we require that $\Gamma, \varphi \vdash u_0 = u(i/0) : A(i/0)$. The result is a term in $A(i/1)$ such that $\mathsf{comp}^i\, A\, [\varphi \mapsto u]\, u_0 = u(i/1)$ on Γ, φ. The computation rules for the composition operations are given as judgmental equalities defined by cases on the type A.

The intuition is that u specifies the sides of an open box while u_0 specifies the bottom of the box and the fact that the sides have to be connected to the bottom is expressed by the equation relating u_0 and $u(i/0)$. The result of the composition operation is then the lid of this open box. For example, given paths p, q, and r as in:

$$\begin{tikzcd}
c \arrow[r, dashed] & d & & i \\
a \arrow[u, "q\ i"] \arrow[r, "p\ j"'] & b \arrow[u, "r\ i"'] & & \cdot \arrow[u] \arrow[r] & j
\end{tikzcd}$$

the composition $\mathsf{comp}^i\, A\, [(j = 0) \mapsto q\ i, (j = 1) \mapsto r\ i]\, (p\ j)$ is the dashed line at the top of the square.[5] Here $p\ j$ is a line in $A(i/0)$ while $q\ i$ and $r\ i$ are lines in $A(j/0)$ and $A(j/1)$, respectively. The resulting composition is then a line in $A(i/1)$.

The composition operations allows us to define transport from a line type:

$$\frac{\Gamma, i : \mathbb{I} \vdash A \qquad \Gamma \vdash u_0 : A(i/0)}{\Gamma \vdash \mathsf{transport}^i\, A\, u_0 = \mathsf{comp}^i\, A\, []\, u_0 : A(i/1)}$$

Combined with "contractibility of singletons" (which is directly provable using a connection) we get the induction principle for Path-types, which means that they behave like Martin-Löf's identity types (modulo the computation rule for the induction principle which only holds up to a Path).

The Glue-types allow us to prove both the univalence axiom and that the universe has a composition operation, however as they do not play an important role in this paper we omit them from this introduction to cubical type theory.

[5] Note that we are using a notation for the "system" $[(i = 0) \mapsto q\ j, (i = 1) \mapsto r\ j]$. Formally this is given by the formula $(i = 0) \vee (i = 1)$ and a partial element with endpoints $q\ j$ and $r\ j$.

3.2 A common pattern for higher inductive types

All of the examples of higher inductive types that we consider in this paper follow a common pattern. In this section we sketch this pattern which can be seen as a first step towards formulating a syntactic schema for higher inductive types in cubical type theory, however the precise formulation of this schema and its semantic counterpart is left as future work.

Each higher inductive type $\mathsf{D}(\vec{z} : \vec{P})$ is specified by a telescope[6] of parameters $\vec{z} : \vec{P}$ (over an ambient context Γ) and a *list* of constructors $\vec{\mathsf{c}}$. Each c in $\vec{\mathsf{c}}$ is specified by the data:

$$\mathsf{c} : (\vec{x} : \vec{A}(\vec{z}))\,[\vec{i}]\,\mathsf{D}(\vec{z})[\varphi(\vec{i}) \mapsto e(\vec{z}, \vec{x}, \vec{i})]$$

Here the telescope $\vec{x} : \vec{A}$ specifies the types of the arguments to c, and in the case of *recursive* higher inductive types, as in, e.g., propositional truncation, D might itself appear in $\vec{A}$. The length of the list of names $\vec{i}$ specifies the dimension of the cube c introduces: we say that c is a *point*, *path*, or *square* constructor according to whether the length of $\vec{i}$ is 0, 1, or 2, respectively. The data $\varphi \mapsto e$ specifies the *endpoints* of the constructor c, with φ an element of the face lattice $\mathbb{F}$ whose free variables are among $\vec{i}$, and e is a partial element

$$\vec{z} : \vec{P}, \vec{x} : \vec{A}(\vec{z}), \vec{i} : \mathbb{I}, \varphi(\vec{i}) \vdash e(\vec{z}, \vec{x}, \vec{i}) : \mathsf{D}(\vec{z})$$

mentioning only previous constructors in the list $\vec{\mathsf{c}}$ and possibly hcomp's (see below).

For each instance $\vec{u} : \vec{P}$ of the telescope $\vec{z} : \vec{P}$ we say that $\mathsf{D}(\vec{u})$ is a type and we will have an introduction rule for a constructor c specified as above

$$\frac{\vec{v} : \vec{A}(\vec{u}) \qquad \vec{r} : \mathbb{I}}{\mathsf{c}\,\vec{v}\,\vec{r} : \mathsf{D}(\vec{u})}$$

and a judgmental equality $\mathsf{c}\,\vec{v}\,\vec{r} = e(\vec{u}, \vec{v}, \vec{r}) : \mathsf{D}(\vec{u})$ in case we additionally have $\varphi(\vec{r}) = 1 : \mathbb{F}$ (all in an ambient context). Note that this judgmental equality for c requires us to make sure that whenever we define a function $\mathsf{f} : \Pi(x : \mathsf{D}(\vec{u}))\,P(x)$ that its semantics preserve this equality, so that

$$\varphi(\vec{r}) \vdash \mathsf{f}(\mathsf{c}\,\vec{v}\,\vec{r}) = \mathsf{f}(e(\vec{u}, \vec{v}, \vec{r})) : P(\mathsf{c}\,\vec{v}\,\vec{r}).$$

In particular, this requirement has to be taken care of in the typing rules for the eliminator for $\mathsf{D}(\vec{u})$. The general formulation of this is left as future work as it would require us to extend cubical type theory with something similar to the "extension types" of [21].

Recall from Section 2 that we decomposed the composition structure for higher inductive types into a homogeneous composition structure and a transport structure. The homogeneous composition structure was introduced as constructors and the same is reflected in the syntax by adding a rule

$$\frac{\begin{array}{c}\Gamma \vdash \vec{u} : \vec{P} \qquad \Gamma \vdash \varphi : \mathbb{F} \\ \Gamma, i : \mathbb{I}, \varphi \vdash v : \mathsf{D}(\vec{u}) \qquad \Gamma \vdash v_0 : \mathsf{D}(\vec{u})[\varphi \mapsto v(i/0)]\end{array}}{\Gamma \vdash \mathsf{hcomp}^i_{\mathsf{D}(\vec{u})}\,[\varphi \mapsto v]\,v_0 : \mathsf{D}(\vec{u})[\varphi \mapsto v(i/1)]}$$

where the key point is that i may be free in v, but *not* in $D(\vec{u})$, as opposed to the composition operations where i may be free in both v and $D(\vec{u})$. In the examples we will not repeat these homogeneous composition constructors for every higher inductive type we consider and they are always assumed to be included as part of the definition of the higher inductive type under consideration.

We could do the same for traditional inductive types like the natural numbers and have a constructor $\mathsf{hcomp}^i_{\mathsf{N}}$ instead of explaining composition by recursion. We can prove that this "weaker" form of natural numbers type is equivalent, and hence equal (by univalence) to the regular one.

To reflect the transport structure in the syntax we specify a trans operation for higher inductive types $A := \mathsf{D}(\vec{u})$ given $\Gamma, i : \mathbb{I} \vdash \vec{u} : \vec{P}$ by the rule:

$$\frac{\Gamma \vdash \varphi : \mathbb{F} \qquad \Gamma, i : \mathbb{I}, \varphi \vdash A = A(i/0) \qquad \Gamma \vdash u_0 : A(i/0)}{\Gamma \vdash \mathsf{trans}^i\,A\,\varphi\,u_0 : A(i/1)[\varphi \mapsto u_0]}$$

Note that since $\Gamma, i : \mathbb{I}, \varphi \vdash A = A(i/0)$ also $\Gamma, \varphi \vdash A(i/0) = A(i/1)$ (and hence this equation also holds in context $\Gamma, i : \mathbb{I}, \varphi$).

Similar to how the transport structure is explained in the semantics by recursion on the argument we will add a judgmental equality for each of the possible shapes of u_0: one for each constructor c and one for the hcomp constructor:

$$\begin{aligned}&\mathsf{trans}^i A\,\varphi\,(\mathsf{hcomp}^j_{A(i/0)}\,[\psi \mapsto u]\,u_0) = \\ &\qquad \mathsf{hcomp}^j_{A(i/1)}\,[\psi \mapsto \mathsf{trans}^i A\,\varphi\,u]\,(\mathsf{trans}^i A\,\varphi\,u_0)\end{aligned}$$

(Note that we can assume that $i \neq j$ as we can always rename one of them as they are both bound.) As the hcomp case is the same for all examples we omit it from the definition of trans for the higher inductive types considered in Section 3.3.

We can define a derived "squeeze" operation analogous to sq_A in the proof of Lemma 2.11:

$$\frac{\Gamma \vdash \varphi : \mathbb{F} \qquad \Gamma, i : \mathbb{I}, \varphi \vdash A = A(i/0) \qquad \Gamma, i : \mathbb{I} \vdash a : A}{\Gamma, i : \mathbb{I} \vdash \mathsf{squeeze}^i\,A\,\varphi\,a := \mathsf{trans}^j\,A(i/i \vee j)\,(\varphi \vee (i = 1))\,a : A(i/1)}$$

This operation satisfies

$$\begin{aligned}(\mathsf{squeeze}^i\,A\,\varphi\,a)(i/0) &= \mathsf{trans}^j\,A(i/j)\,\varphi\,a(i/0) \\ (\mathsf{squeeze}^i\,A\,\varphi\,a)(i/1) &= a(i/1)\end{aligned}$$

and the induced path is constantly a on φ.

Assuming that we have defined trans for a higher inductive type $\Gamma, i : \mathbb{I} \vdash A$ we can define the composition operation:

$$\frac{\Gamma \vdash \varphi : \mathbb{F} \qquad \Gamma, i : \mathbb{I}, \varphi \vdash u : A \qquad \Gamma \vdash u_0 : A(i/0)[\varphi \mapsto u(i/0)]}{\begin{array}{c}\Gamma \vdash \mathsf{comp}^i\,A\,[\varphi \mapsto u]\,u_0 := \\ \mathsf{hcomp}^i_{A(i/1)}\,[\varphi \mapsto \mathsf{squeeze}^i\,A\,0_{\mathbb{F}}\,u]\,(\mathsf{trans}^i\,A\,0_{\mathbb{F}}\,u_0) : A(i/1)\end{array}}$$

This satisfies the required judgmental computation rule for comp because of the computation rules for hcomp and $\mathsf{squeeze}$. This means that in order to define the composition operation for a higher inductive type we only need to define the trans operation when applied to constructors.

Note, that we can always define a trans operation for any type $\Gamma, i : \mathbb{I} \vdash A$ that already has a composition operation by:

$$\frac{\Gamma \vdash \varphi : \mathbb{F} \qquad \Gamma, i : \mathbb{I}, \varphi \vdash A = A(i/0) \qquad \Gamma \vdash u_0 : A(i/0)}{\Gamma \vdash \mathsf{ctrans}^i\,A\,\varphi\,u_0 := \mathsf{comp}^i\,A\,[\varphi \mapsto u_0]\,u_0 : A(i/1)[\varphi \mapsto u_0]}$$

[6] A *telescope* $x_1 : A_1, \ldots, x_n : A_n$ (written as $\vec{x} : \vec{A}$) over a context Γ is a (possibly empty) list of object variable declarations such that $\Gamma, \vec{x} : \vec{A}$ is a well-formed context, so $\vec{x} : \vec{A}$ neither contains context restrictions Δ, φ nor dimension variables $i : \mathbb{I}$.

In line with Lemma 2.11 a corresponding "filling" operation which connects the input of trans to its output can also be derived:

$$\frac{\Gamma \vdash \varphi : \mathbb{F} \qquad \Gamma, i : \mathbb{I}, \varphi \vdash A = A(i/0) \qquad \Gamma \vdash u_0 : A(i/0)}{\Gamma, i : \mathbb{I} \vdash \mathsf{transFill}^i\, A\, \varphi\, u_0 := \mathsf{trans}^j\, A(i/i \wedge j)\, (\varphi \vee (i = 0))\, u_0 : A}$$

Note that $\Gamma, i : \mathbb{I}, \varphi \vdash A = A(i/0)$ entails

$$\Gamma, i : \mathbb{I}, j : \mathbb{I}, \varphi \vee (i = 0) \vdash A(i/i \wedge j) = A(i/i \wedge j)(j/0).$$

This operation satisfies

$$(\mathsf{transFill}^i\, A\, \varphi\, u_0)(i/0) = u_0$$
$$(\mathsf{transFill}^i\, A\, \varphi\, u_0)(i/1) = \mathsf{trans}^j\, A(i/j)\, \varphi\, u_0$$

and the induced path is constantly u_0 on φ. We write ctransFill for the corresponding operation defined using ctrans.

3.3 Examples of higher inductive types

In this section we describe how to extend cubical type theory with the circle and spheres, torus, suspensions, propositional truncation, and pushouts. As with all the other type formers we have to explain their formation, introduction, elimination, and computation rules, as well as how composition computes. All of these examples follow the common pattern presented in the previous section.

3.3.1 The circle and spheres

The extension of cubical type theory with the circle and spheres was sketched in [8, Section 9.2] and we elaborate on this here.

Formation In order to extend the theory with the circle we first add it as a type by:

$$\frac{}{\vdash \mathbb{S}^1} \qquad \frac{}{\mathbb{S}^1 : \mathsf{U}}$$

Introduction The circle is generated by a point and a path constructor:

$$\frac{}{\mathsf{base} : \mathbb{S}^1} \qquad \frac{r : \mathbb{I}}{\mathsf{loop}\, r : \mathbb{S}^1}$$

with the judgmental equalities $\mathsf{loop}\, 0 = \mathsf{loop}\, 1 = \mathsf{base}$ so that loop connects the point to itself.

Elimination Given a dependent type $x : \mathbb{S}^1 \vdash P(x)$, a term $b : P(\mathsf{base})$ and a path $i : \mathbb{I} \vdash l : P(\mathsf{loop}\, i)[(i = 0) \vee (i = 1) \mapsto b]$ we can define $\mathsf{f} : \Pi(x : \mathbb{S}^1)\, P(x)$ by cases:

$$\mathsf{f}\, \mathsf{base} = b \qquad \mathsf{f}\, (\mathsf{loop}\, r) = l\, r$$

and for the hcomp constructor:

$$\mathsf{f}\, (\mathsf{hcomp}^i_{\mathbb{S}^1}\, [\varphi \mapsto u]\, u_0) = \mathsf{comp}^i\, P(v)\, [\varphi \mapsto \mathsf{f}\, u]\, (\mathsf{f}\, u_0)$$

where w.l.o.g. i is fresh and:

$$\begin{aligned} v &:= \mathsf{hfill}^i\, \mathbb{S}^1\, [\varphi \mapsto u]\, u_0 \\ &= \mathsf{hcomp}^j_{\mathbb{S}^1} [\varphi \mapsto u(i/i \wedge j), (i = 0) \mapsto u_0]\, u_0 \end{aligned}$$

As the equation for the eliminator applied to an hcomp is analogous for all the other higher inductive considered here we will omit it in the sequel.

Using this we can directly define the eliminator:

$$\frac{x : \mathbb{S}^1 \vdash P(x) \qquad b : P(\mathsf{base}) \qquad l : \mathsf{Path}^i\, P(\mathsf{loop}\, i)\, b\, b \qquad u : \mathbb{S}^1}{\mathbb{S}^1\text{-}\mathsf{elim}_{x.P}\ b\, l\, u : P(u)}$$

where Path^i denotes a dependent path type (see [8, Section 9.2]). The judgmental computation rules then follow from the definition above. Note that as we have dependent Path-types (which behave like heterogeneous equalities) the loop case of f can be expressed directly by an equation without "apd" and l does not involve any transport as opposed to [26, Section 6.4].

Composition As $\mathbb{S}^1$ has no parameters we let $\mathsf{trans}^i\, \mathbb{S}^1\, \varphi\, u_0 = u_0$. This means that the composition $\mathsf{comp}^i\, \mathbb{S}^1\, [\varphi \mapsto u]\, u_0$ computes directly to the constructor $\mathsf{hcomp}^i_{\mathbb{S}^1}\, [\varphi \mapsto u]\, u_0$.

The higher dimensional spheres, $\mathbb{S}^n$, can directly be defined by generalizing the definition $\mathbb{S}^1$ so that loop takes $r_1, \dots, r_n : \mathbb{I}$. It is trivial to define $\mathsf{trans}^i\, \mathbb{S}^n\, \varphi\, u_0$ in analogy with $\mathbb{S}^1$. The elimination is also analogous to that of $\mathbb{S}^1$ using an n-dimensional cube in $P(\mathsf{loop}\, i_1\, \dots\, i_n)$ for the loop case.

3.3.2 The torus; two equivalent formulations

We define the torus in two ways, the first one (written $\mathbb{T}$) is analogous to $\mathbb{S}^2$ and the second (written $\mathbb{T}_\mathsf{F}$) is the cubical analogue of the torus as defined in [26, Section 6.6]. The $\mathbb{T}_\mathsf{F}$ torus involves the fibrancy structure of the 1-dimensional cells in the 2-dimensional cell. Higher inductive types of this kind are not supported by [19] and we make crucial use of the fact that we have homogeneous composition as a constructor in order to represent them.

Formation The formation rules for the torus types are given by:

$$\frac{}{\vdash \mathbb{T}} \qquad \frac{}{\mathbb{T} : \mathsf{U}} \qquad \frac{}{\vdash \mathbb{T}_\mathsf{F}} \qquad \frac{}{\mathbb{T}_\mathsf{F} : \mathsf{U}}$$

Introduction The point, lines and square constructors for $\mathbb{T}$ are given by:

$$\frac{}{\mathsf{b} : \mathbb{T}} \qquad \frac{r : \mathbb{I}}{\mathsf{tp}\, r : \mathbb{T}} \qquad \frac{r : \mathbb{I}}{\mathsf{tq}\, r : \mathbb{T}} \qquad \frac{r : \mathbb{I} \qquad s : \mathbb{I}}{\mathsf{surf}\, r\, s : \mathbb{T}}$$

satisfying $\mathsf{tp}\, 0 = \mathsf{tp}\, 1 = \mathsf{tq}\, 0 = \mathsf{tq}\, 1 = \mathsf{b}$. The constructors for $\mathbb{T}_\mathsf{F}$ are defined by the same rules as for $\mathbb{T}$ and we write them with F as subscript. The square constructor for $\mathbb{T}$ satisfies $\mathsf{surf}\, 0\, s = \mathsf{surf}\, 1\, s = \mathsf{tp}\, s$ and $\mathsf{surf}\, r\, 0 = \mathsf{surf}\, r\, 1 = \mathsf{tq}\, r$ so that we get the square representing the traditional gluing diagram used in the topological definition of the torus:

$$\begin{tikzcd} \mathsf{b} \arrow[r, "\mathsf{tq}\, i"] & \mathsf{b} \\ \mathsf{b} \arrow[u, "\mathsf{tp}\, j"] \arrow[r, "\mathsf{tq}\, i"'] \arrow[ur, phantom, "\mathsf{surf}\, i\, j"] & \mathsf{b} \arrow[u, "\mathsf{tp}\, j"'] \end{tikzcd} \qquad \begin{tikzcd} {} \\ \cdot \arrow[u, "j"] \arrow[r, "i"'] & {} \end{tikzcd}$$

Given $s : \mathbb{I}$ we define the composition of tp_F and tq_F by:

$$\mathsf{tp}_\mathsf{F} \cdot_s \mathsf{tq}_\mathsf{F} := \mathsf{hcomp}^i_{\mathbb{T}_\mathsf{F}}\, [(s = 0) \mapsto \mathsf{b}_\mathsf{F}, (s = 1) \mapsto \mathsf{tq}_\mathsf{F}\, i]\, (\mathsf{tp}_\mathsf{F}\, s)$$

The composition $\mathsf{tq}_\mathsf{F} \cdot_s \mathsf{tp}_\mathsf{F}$ is defined analogously.

The square constructor for $\mathbb{T}_\mathsf{F}$ satisfies $\mathsf{surf}_\mathsf{F}\, 0\, s = \mathsf{tp}_\mathsf{F} \cdot_s \mathsf{tq}_\mathsf{F}$, $\mathsf{surf}_\mathsf{F}\, 1\, s = \mathsf{tq}_\mathsf{F} \cdot_s \mathsf{tp}_\mathsf{F}$ and $\mathsf{surf}_\mathsf{F}\, r\, 0 = \mathsf{surf}_\mathsf{F}\, r\, 1 = \mathsf{b}_\mathsf{F}$. This way the 2-cell $\langle i\, j \rangle\, \mathsf{surf}_\mathsf{F}\, i\, j$ corresponds to a cubical version of the globe (which can be turned into a square with reflexivity at b_F as sides):

$$\begin{tikzcd} \mathsf{b}_\mathsf{F} \arrow[r, bend left, "\mathsf{tp}_\mathsf{F} \cdot_j \mathsf{tq}_\mathsf{F}"] \arrow[r, bend right, "\mathsf{tq}_\mathsf{F} \cdot_j \mathsf{tp}_\mathsf{F}"'] & \mathsf{b}_\mathsf{F} \end{tikzcd}$$

Elimination We write $(i = 0/1)$ for $(i = 0) \vee (i = 1)$. Given a dependent type $x : \mathbb{T} \vdash P(x)$, a term $b : P(\mathsf{b})$, paths $i : \mathbb{I} \vdash l_p : P(\mathsf{tp}\, i)[(i = 0/1) \mapsto b]$ and $i : \mathbb{I} \vdash l_q : P(\mathsf{tq}\, i)[(i = 0/1) \mapsto b]$ and a square $i, j : \mathbb{I} \vdash s_{pq} : P(\mathsf{surf}\, i\, j)\,[(i = 0/1) \mapsto l_p\, j, (j = 0/1) \mapsto l_q\, i]$ we can define $\mathsf{f} : \Pi(x : \mathbb{T})\, P(x)$ by cases:

$$\mathsf{f}\,\mathsf{b} = b \qquad \mathsf{f}\,(\mathsf{tp}\, r) = l_p\, r \qquad \mathsf{f}\,(\mathsf{tq}\, r) = l_q\, r \qquad \mathsf{f}\,(\mathsf{surf}\, r\, s) = s_{pq}\, r\, s$$

Similarly for a dependent type $x : \mathbb{T}_\mathsf{F} \vdash P(x)$, a term $b : P(\mathsf{b}_\mathsf{F})$, paths $i : \mathbb{I} \vdash l_p : P(\mathsf{tp}_\mathsf{F}\, i)[(i = 0/1) \mapsto b]$ and $i : \mathbb{I} \vdash l_q : P(\mathsf{tq}_\mathsf{F}\, i)[(i = 0/1) \mapsto b]$ we define:

$$l_p \cdot_j l_q := \mathsf{comp}^i\, P(v)\, [(j = 0) \mapsto b, (j = 1) \mapsto l_q\, i]\, (l_p\, j)$$

where $v := \mathsf{hfill}^i_{\mathbb{T}_\mathsf{F}}\, [(j = 0) \mapsto \mathsf{b}_\mathsf{F}, (j = 1) \mapsto \mathsf{tq}_\mathsf{F}\, i]\, (\mathsf{tp}_\mathsf{F}\, j)$. We define $l_q \cdot_j l_p$ analogously and we can then require a square $i, j : \mathbb{I} \vdash s_{pq} : P(\mathsf{surf}_\mathsf{F}\, i\, j)\, [(i = 0) \mapsto l_p \cdot_j l_q, (i = 1) \mapsto l_q \cdot_j l_p, (j = 0/1) \mapsto b]$. Using this we can define $\mathsf{f} : \Pi(x : \mathbb{T}_\mathsf{F})\, P(x)$ by cases like for $\mathbb{T}$.

Working with $\mathbb{T}$ is easier than $\mathbb{T}_\mathsf{F}$ and the proof that $\mathbb{T} \simeq \mathbb{S}^1 \times \mathbb{S}^1$ has been formalized in CUBICALTT by Dan Licata.[7] The proof of this is very direct and a lot shorter than the existing proofs in the literature [17, 24]. One first defines maps $f_1 : \mathbb{T} \to \mathbb{S}^1 \times \mathbb{S}^1$ and $f_2 : \mathbb{S}^1 \times \mathbb{S}^1 \to \mathbb{T}$ by:

$$\begin{aligned} f_1\, \mathsf{b} &= (\mathsf{base}, \mathsf{base}) & f_2\, (\mathsf{base}, \mathsf{base}) &= \mathsf{b} \\ f_1\, (\mathsf{tp}\, r) &= (\mathsf{loop}\, r, \mathsf{base}) & f_2\, (\mathsf{loop}\, r, \mathsf{base}) &= \mathsf{tp}\, r \\ f_1\, (\mathsf{tq}\, r) &= (\mathsf{base}, \mathsf{loop}\, r) & f_2\, (\mathsf{base}, \mathsf{loop}\, r) &= \mathsf{tq}\, r \\ f_1\, (\mathsf{surf}\, r\, s) &= (\mathsf{loop}\, r, \mathsf{loop}\, s) & f_2\, (\mathsf{loop}\, r, \mathsf{loop}\, s) &= \mathsf{surf}\, r\, s \end{aligned}$$

These are obviously inverses and the equivalence can be established. The formal proof in CUBICALTT is slightly more complicated as it is not possible to directly do the double recursion in f_2, but the basic idea is the same. This example shows how having a system where higher inductive types compute also for higher constructors makes it possible to simplify formal proofs in synthetic homotopy theory.

Composition As neither $\mathbb{T}$ or $\mathbb{T}_\mathsf{F}$ have any parameters the transport operation is trivial just like for $\mathbb{S}^n$, so the composition operations reduces to the hcomp constructors.

3.3.3 Suspension

The suspension of a type A, written $\mathsf{Susp}\, A$, is more involved than the higher inductive types considered so far as it has a parameter and just as in the semantics we have to explain the transport operation.

Formation In order to extend the theory with suspensions we add the rules:

$$\frac{\vdash A}{\vdash \mathsf{Susp}\, A} \qquad \frac{A : \mathsf{U}}{\mathsf{Susp}\, A : \mathsf{U}}$$

Note that we allow $\mathsf{Susp}\, A$ to be in the same universe as A, this is justified by the semantics as explained in Section 2.5.

Introduction The suspensions are generated by:

$$\frac{}{\mathsf{N} : \mathsf{Susp}\, A} \qquad \frac{}{\mathsf{S} : \mathsf{Susp}\, A} \qquad \frac{a : A \qquad r : \mathbb{I}}{\mathsf{merid}\, a\, r : \mathsf{Susp}\, A}$$

satisfying $\mathsf{merid}\, a\, 0 = \mathsf{N}$ and $\mathsf{merid}\, a\, 1 = \mathsf{S}$.

Elimination Given a dependent type $x : \mathsf{Susp}\, A \vdash P(x)$, terms $n : P(\mathsf{N})$ and $s : P(\mathsf{S})$ and a family of paths $x : A, i : \mathbb{I} \vdash m(x, i) : P(\mathsf{merid}\, x\, i)[(i = 0) \mapsto n, (i = 1) \mapsto s]$ we can define a function $\mathsf{f} : \Pi(x : \mathsf{Susp}\, A)\, P(x)$ by cases:

$$\mathsf{f}\, \mathsf{N} = n \qquad \mathsf{f}\, \mathsf{S} = s \qquad \mathsf{f}\, (\mathsf{merid}\, a\, r) = m(a, r)$$

Composition The $\mathsf{trans}^i (\mathsf{Susp}\, A)\, \varphi\, u_0$ operation is defined as

$$\mathsf{trans}^i (\mathsf{Susp}\, A)\, \varphi\, \mathsf{N} = \mathsf{N} \qquad \mathsf{trans}^i (\mathsf{Susp}\, A)\, \varphi\, \mathsf{S} = \mathsf{S}$$

$$\mathsf{trans}^i (\mathsf{Susp}\, A)\, \varphi\, (\mathsf{merid}\, a\, r) = \mathsf{merid}\, (\mathsf{ctrans}^i\, A\, \varphi\, a)\, r$$

3.3.4 Propositional truncations

Another class of interesting higher inductive types are the truncations; these introduce some new complications as they are recursive in the sense that the higher constructors quantify over elements of the type. The propositional truncation takes a type A and "squashes" it to a 0-type $\|A\|$ (in the sense that the equality type of $\|A\|$ has no interesting structure).

Formation In order to extend the theory with propositional truncation we add the rules:

$$\frac{\vdash A}{\vdash \|A\|} \qquad \frac{A : \mathsf{U}}{\|A\| : \mathsf{U}}$$

Introduction The propositional truncation of A is generated by:

$$\frac{a : A}{\mathsf{inc}\, a : \|A\|} \qquad \frac{v : \|A\| \qquad w : \|A\| \qquad r : \mathbb{I}}{\mathsf{sq}\, v\, w\, r : \|A\|}$$

satisfying $\mathsf{sq}\, v\, w\, 0 = v$ and $\mathsf{sq}\, v\, w\, 1 = w$.

Elimination Given a dependent type $x : \|A\| \vdash P(x)$, a family of terms $x : A \vdash t(x) : P(\mathsf{inc}\ x)$ and family of paths $v, w : \|A\|, x : P(v), y : P(w), i : \mathbb{I} \vdash p(v, w, x, y, i) : P(\mathsf{sq}\, v\, w\, i)[(i = 0) \mapsto x, (i = 1) \mapsto y]$ we can define $\mathsf{f} : \Pi(x : \|A\|)\, P(x)$ by cases:

$$\mathsf{f}\, (\mathsf{inc}\, a) = t(a) \qquad \mathsf{f}\, (\mathsf{sq}\, v\, w\, r) = p(v, w, \mathsf{f}\, v, \mathsf{f}\, w, r)$$

This is directly structurally recursive and the only difference compared to $\mathsf{Susp}\, A$ is that we have to make a recursive call for each recursive argument.

Composition We define $\mathsf{trans}^i \|A\|\, \varphi\, u_0$ by cases on u_0:

$$\mathsf{trans}^i \|A\|\, \varphi\, (\mathsf{inc}\ a) = \mathsf{inc}\, (\mathsf{ctrans}^i\, A\, \varphi\, a)$$

$$\mathsf{trans}^i \|A\|\, \varphi\, (\mathsf{sq}\, v\, w\, r) = \mathsf{sq}\, (\mathsf{trans}^i \|A\|\, \varphi\, v)\, (\mathsf{trans}^i \|A\|\, \varphi\, w)\, r$$

The explanation of propositional truncation in [8, Section 9.2] used a similar decomposition, but the introduction of the trans operation allows a much simpler formulation of composition.

3.3.5 Pushouts

The definition of pushouts in cubical type theory is similar to the other parametrized higher inductive types, but special care has to be taken when defining trans as the endpoints of the path constructors involve the parameters to the pushout.

Formation We extend the theory with:

$$\frac{\vdash A \qquad \vdash B \qquad \vdash C \qquad u : C \to A \qquad v : C \to B}{\vdash A \sqcup_C B}$$

$$\frac{A : \mathsf{U} \qquad B : \mathsf{U} \qquad C : \mathsf{U} \qquad u : C \to A \qquad v : C \to B}{A \sqcup_C B : \mathsf{U}}$$

[7] See: https://github.com/mortberg/cubicaltt/blob/hcomptrans/examples/torus.ctt

Introduction Given $u : C \to A$ and $v : C \to B$ the pushout is generated by:

$$\frac{a : A}{\mathsf{inl}\, a : A \sqcup_C B} \qquad \frac{b : B}{\mathsf{inr}\, b : A \sqcup_C B} \qquad \frac{c : C \qquad r : \mathbb{I}}{\mathsf{push}\, c\, r : A \sqcup_C B}$$

satisfying $\mathsf{push}\, c\, 0 = \mathsf{inl}(u\, c)$ and $\mathsf{push}\, c\, 1 = \mathsf{inr}(v\, c)$. Note that $\langle i \rangle\, \mathsf{push}\, c\, i$ gives a path between $\mathsf{inl}(u\, c)$ and $\mathsf{inr}(v\, c)$ for all $c : C$ as desired.

Elimination Given a dependent type $x : A \sqcup_C B \vdash P(x)$, families of terms $x : A \vdash l(x) : P(\mathsf{inl}\ x)$ and $x : B \vdash r(x) : P(\mathsf{inr}\ x)$ and a family of paths $x : C, i : \mathbb{I} \vdash p(x, i) : P(\mathsf{push}\, x\, i)[(i = 0) \mapsto l(u\, x), (i = 1) \mapsto r(v\, x)]$ we can define $\mathsf{f} : \Pi(x : A \sqcup_C B)\, P(x)$ by cases:

$$\mathsf{f}\,(\mathsf{inl}\, a) = l(a) \qquad \mathsf{f}\,(\mathsf{inr}\, b) = r(b) \qquad \mathsf{f}\,(\mathsf{push}\, c\, r) = p(c, r)$$

Composition We write P for $A \sqcup_C B$ and the judgmental computation rules for trans are defined by cases:

$$\mathsf{trans}^i\, P\, \varphi\, (\mathsf{inl}\, a) = \mathsf{inl}\, (\mathsf{ctrans}^i\, A\, \varphi\ a)$$
$$\mathsf{trans}^i\, P\, \varphi\, (\mathsf{inr}\, b) = \mathsf{inr}\, (\mathsf{ctrans}^i\, B\, \varphi\ b)$$
$$\mathsf{trans}^i\, P\, \varphi\, (\mathsf{push}\, c\, r) = \mathsf{hcomp}^i_{P(i/1)}\, S\, (\mathsf{push}\, (\mathsf{ctrans}^i\, C\, \varphi\, c)\, r)$$

where S is the system:

$$[(r = 0) \mapsto \mathsf{squeeze}^i\, P\, \varphi\, (\mathsf{inl}(u\, (\mathsf{ctransFill}^i\, C\, \varphi\, c)))\, (i/1 - i),$$
$$(r = 1) \mapsto \mathsf{squeeze}^i\, P\, \varphi\, (\mathsf{inr}(v\, (\mathsf{ctransFill}^i\, C\, \varphi\, c)))\, (i/1 - i),$$
$$(\varphi = 1) \mapsto \mathsf{push}\, c\, r]$$

Note that the recursive call to squeeze is justified as it is applied to a point constructor which has already been defined.

Furthermore, note that the endpoint correction for $\mathsf{push}\, c\, r$ is necessary as, for example, in the case where r is a dimension variable j the path constructor $\mathsf{push}\, (\mathsf{ctrans}^i\, C\, \varphi\, c)\, j$ connects

$$\mathsf{inl}(u(i/1)\, (\mathsf{ctrans}^i\, C\, \varphi\, c)) \qquad \text{to} \qquad \mathsf{inr}(v(i/1)\, (\mathsf{ctrans}^i\, C\, \varphi\, c))$$

in direction j, but we require something that connects

$$\mathsf{inl}(\mathsf{ctrans}^i\, A\, \varphi\, (u(i/0)\, c)) \qquad \text{to} \qquad \mathsf{inr}(\mathsf{ctrans}^i\, B\, \varphi\, (v(i/0)\, c))$$

since the definition of trans should be stable under the substitutions $(j/0)$ and $(j/1)$. To see that the correction is correct at $(r = 0)$ note that $\mathsf{squeeze}^i\, P\, \varphi\, (\mathsf{inl}(u\, (\mathsf{ctransFill}^i\, C\, \varphi\, c)))\, (i/1 - i)$ connects

$$\mathsf{inl}(u(i/1)\, (\mathsf{ctrans}^i\, C\, \varphi\, c)) \qquad \text{to} \qquad \mathsf{inl}(\mathsf{ctrans}^i\, A\, \varphi\, (u(i/0)\, c))$$

as required.

3.4 A variation on cubical type theory

In the previous section we have seen that the equations to define $\mathsf{trans}^i\, A$ for a higher inductive type A applied to a constructor involves $\mathsf{trans}^i\, A$ for the recursive arguments to the constructor (see the equation for $\mathsf{sq}\, v\, w\, r$ for propositional truncation in Section 3.3.4), and involves the derived operations ctrans for non-recursive arguments (e.g., in the equation for $\mathsf{merid}\, a\, r$ in Section 3.3.3). In general, trans and ctrans which are available for A do not coincide definitionally, making it impossible to treat the recursive and non-recursive arguments to a constructor uniformly.

This mismatch suggests a variant of cubical type theory where the operations trans and hcomp are taken as primitives and comp is instead a derived operation as we did here for higher inductive types. We can then explain trans and hcomp by cases on the shape of the type. In this variation of cubical type theory the algorithm for trans in a higher inductive type applied to a constructor can be uniformly described as follows.

Given a higher inductive type $\mathsf{D}(\vec{z} : \vec{P})$ specified as in Section 3.2 and a constructor c specified by:

$$\mathsf{c} : (\vec{x} : \vec{A}(\vec{z}))\, [\vec{i}]\, \mathsf{D}(\vec{z})[\varphi(\vec{i}) \mapsto e(\vec{z}, \vec{x}, \vec{i})] \tag{1}$$

Further, assume parameters $\Gamma, i : \mathbb{I} \vdash \vec{u} : \vec{P}$ of the higher inductive type $\mathsf{D}(\vec{z} : \vec{P})$ such that $\Gamma, i : \mathbb{I}, \psi \vdash \vec{u} = \vec{u}(i/0) : \vec{P}$ for $\Gamma \vdash \psi : \mathbb{F}$. We now explain the judgmental equalities of

$$w_1 := \mathsf{trans}^i\, \mathsf{D}(\vec{u})\, \psi\, (\mathsf{c}\, \vec{v}\, \vec{r})$$

for $\Gamma \vdash \vec{v} : \vec{A}(\vec{u}(i/0))$ and $\Gamma \vdash \vec{r} : \mathbb{I}$. This $\mathsf{c}\, \vec{v}\, \vec{r}$ restricts to $\varphi(\vec{r}) \mapsto e(\vec{u}(i/0), \vec{v}, \vec{r})$. We want to define $\Gamma \vdash w_1 : \mathsf{D}(\vec{u}(i/1))[\psi \mapsto \mathsf{c}\, \vec{v}\, \vec{r}]$ such that w_1 restricts to

$$\varphi(\vec{r}) \mapsto \mathsf{trans}^i\, \mathsf{D}(\vec{u})\, \psi\, e(\vec{u}(i/0), \vec{v}, \vec{r}). \tag{2}$$

We get a line in $\vec{x} : \vec{A}(\vec{u})$ in the context $\Gamma, i : \mathbb{I}$

$$\vec{v} \xrightarrow{\ \vec{\theta} := \mathsf{transFill}^i\, (\vec{x} : \vec{A}(\vec{u}))\, \psi\, \vec{v}\ } \mathsf{trans}^i\, (\vec{x} : \vec{A}(\vec{u}))\, \psi\, \vec{v}$$

along i, where $\mathsf{transFill}^i\, (\vec{x} : \vec{A})\, \psi\, \vec{v}$ is the extension of transFill to telescopes, mapping the empty telescope to itself, and

$$\mathsf{transFill}^i\, (x : A, \vec{x} : \vec{A}(x))\, \psi\, (v, \vec{v}) = \tilde{v}, \mathsf{transFill}^i\, (\vec{x} : \vec{A}(\tilde{v}))\, \psi\, \vec{v}$$

with $\tilde{v} = \mathsf{transFill}^i\, A\, \psi\, v$. The extension of trans to telescopes is the $(i/1)$ face of the corresponding transFill.

We start with $\Gamma \vdash w_1' : \mathsf{D}(\vec{u}(i/1))$ given by

$$w_1' := \mathsf{c}\, (\mathsf{trans}^i\, (\vec{x} : \vec{A})\, \psi\, \vec{v})\, \vec{r}$$

which restricts to $\varphi(\vec{r}) \mapsto e(\vec{u}(i/1), \mathsf{trans}^i\, (\vec{x} : \vec{A})\, \psi\, \vec{v}, \vec{r})$ and which we have to correct to match (2). To make this correction, consider the line $\Gamma, \varphi(\vec{r}), i : \mathbb{I} \vdash \alpha(i) : \mathsf{D}(\vec{u}(i/1))$ given by

$$\alpha(i) := \mathsf{squeeze}^i\, \mathsf{D}(\vec{u})\, \psi\, e(\vec{u}, \vec{\theta}, \vec{r})$$

connecting the element in (2) to $e(\vec{u}(i/1), \mathsf{trans}^i\, (\vec{x} : \vec{A})\, \psi\, \vec{v}, \vec{r})$. Note that $\alpha(i)$ coincides with $e(\vec{u}(i/0), \vec{v}, \vec{r})$ (and hence with $\mathsf{c}\, \vec{v}\, \vec{r}$) on ψ.

We now add the judgmental equality

$$w_1 = \mathsf{hcomp}^i_{\mathsf{D}(\vec{u}(i/1))}\, [\varphi(\vec{r}) \mapsto \alpha(1 - i), \psi \mapsto \mathsf{c}\, \vec{v}\, rs]\, w_1'.$$

Note that in the definition α we recursively call trans for D on e. To ensure that this call is well-founded it is crucial to have restrictions on how e may look like.

Also note that this algorithm might not be optimal: For a higher inductive type without any parameters (e.g., $\mathbb{S}^1$) we could have simply defined trans to be the identity as we did in the previous section. For a type where the endpoints of constructors are suitably simple, like suspensions and propositional truncation, but not pushouts, we could have directly taken w_1' above. This has the consequence that the result might have some unnecessary hcomp's and would equal, up to a Path, to a simpler term without these hcomp's.

Our general pattern of constructors (1) suggests to formulate a schema. Such a schema would have to ensure that $\mathsf{D}(\vec{z})$ only appears strictly positive in $\vec{A}$ and would have to restrict what possible endpoints e are allowed. We leave the detailed formulation of the semantics of such a schema as future work.

4 Conclusions and related work

In this paper we constructed the semantics of some important higher inductive types in cubical sets. A crucial ingredient was the decomposition of the composition structure into a homogeneous composition structure and a transport structure. Using this decomposition we define higher inductive type formers such that they preserve the universe level and are strictly stable under substitution.

We also extended cubical type theory with some higher inductive types. While [14] only proves canonicity for cubical type theory extended with the circle and propositional truncation, it should be straightforward to extend this result to the higher inductive types presented in this paper using the obvious operational semantics obtained by orienting the judgmental equalities given here. It also remains to prove normalization and decidability of type-checking for cubical type theory and in particular also for our extension with higher inductive types.

As mentioned in Section 3.4, it is more natural for a general treatment of higher inductive types to formulate a variation of cubical type theory based homogeneous compositions and transport as primitive instead of heterogeneous compositions. It seems that our description of transport for higher inductive types also works for a more general schema, but its details and semantics still have to be worked out.

Using the experimental implementation of the system presented in this paper we have formalized the "Brunerie number"[8], i.e., n such that $\pi_4(\mathbb{S}^3) \simeq \mathbb{Z}/n\mathbb{Z}$. The formalization closely follows [6, Appendix B] and the definition involves multiple higher inductive types (the spheres, truncations, and join construction) together with many uses of the univalence axiom. By the classical definition of this homotopy group we know that the expected value for n is ± 2 and this also is proved to be the case in [6]. But as we have a constructive justification for all of the notions involved in the definition we can in principle directly obtain this numeral by computation. However, this computation so far has been unfeasible.

Further future work is to relate our semantics to other models of homotopy type theory. In particular, clarify the connection of the model structure on cubical sets [23] and the usual model structure on simplicial sets. It is also of interest to investigate to what extent the techniques developed in this paper can be adapted to the simplicial set model.[9]

Related work The papers [2–4, 7] present cubical type theories inspired by an alternative cubical set category with different fibrancy structure, but with the same decomposition of the composition operation in a homogeneous composition and a transport operation. This decomposition was introduced in an early version of [8] precisely to solve the problem of the interpretation of higher inductive types with parameters. The suspensions are covered in [2], and [7] defines a schema for higher inductive types formulated in this setting. The papers [3, 4, 7] describe computational type theories in the style of Nuprl with a semantics where types are interpreted as partial equivalence relations which gives canonicity for booleans. The schema presented in [7] covers all of the examples of higher inductive types considered in this paper.

The paper [19] presents a semantics of higher inductive types in a general framework of "sufficiently nice" Quillen model categories. However as it is now, it models a type theory which does not contain any universes (see [19, pp. 5–6] for a discussion of this point).

A schema with point, path, and square constructors expressed in the style of [26] is presented in [10]. This paper also contains a semantics for these higher inductive types in the groupoid model.

[8]The complete self-contained formalization can be found at: https://github.com/mortberg/cubicaltt/blob/hcomptrans/examples/brunerie.ctt

[9]See the following discussion for more details: https://groups.google.com/d/msg/homotopytypetheory/bNHRnGiF5R4/3RYz1YFmBQAJ

References

[1] Peter Aczel. 1999. On Relating Type Theories and Set Theories. In *Types for Proofs and Programs*. LNCS, Vol. 1657. Springer, 1–18.

[2] Carlo Angiuli, Guillaume Brunerie, Thierry Coquand, Kuen-Bang Hou (Favonia), Robert Harper, and Daniel R. Licata. 2017. Cartesian Cubical Type Theory. (2017). Draft available at https://www.cs.cmu.edu/~rwh/papers/uniform/uniform.pdf.

[3] Carlo Angiuli, Kuen-Bang Hou (Favonia), and Robert Harper. 2017. Computational Higher Type Theory III: Univalent Universes and Exact Equality. (2017). Preprint arXiv:1712.01800v1.

[4] Carlo Angiuli, Robert Harper, and Todd Wilson. 2017. Computational Higher-dimensional Type Theory. In *POPL '17: Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages*. ACM, 680–693.

[5] Steve Awodey and Michael A. Warren. 2009. Homotopy theoretic models of identity types. *Math. Proc. Cambridge Philos. Soc.* 146, 1 (2009), 45–55.

[6] Guillaume Brunerie. 2016. *On the homotopy groups of spheres in homotopy type theory*. Ph.D. Dissertation. Université de Nice.

[7] Evan Cavallo and Robert Harper. 2018. Computational Higher Type Theory IV: Inductive Types. (2018). Preprint arXiv:1801.01568v1.

[8] Cyril Cohen, Thierry Coquand, Simon Huber, and Anders Mörtberg. 2018. Cubical Type Theory: A Constructive Interpretation of the Univalence Axiom. In *Types for Proofs and Programs (TYPES 2015) (LIPIcs)*, Vol. 69. 5:1–5:34.

[9] Floris van Doorn. 2016. Constructing the Propositional Truncation Using Non-recursive HITs. In *CPP '16: Proceedings of the 5th ACM SIGPLAN Conference on Certified Programs and Proofs*. ACM, 122–129.

[10] Peter Dybjer and Hugo Moeneclaey. 2018. Finitary Higher Inductive Types in the Groupoid Model. *Electronic Notes in Theoretical Computer Science* 336 (April 2018), 119–134.

[11] Samuel Eilenberg. 1939. On the relation between the fundamental group on a space and the higher homotopy groups. *Fund. Math.* 32, 1 (1939), 167–175.

[12] Richard Garner. 2009. Understanding the small object argument. *Applied Categorical Structures* 17, 3 (2009), 247–285.

[13] Kuen-Bang Hou (Favonia), Eric Finster, Daniel R. Licata, and Peter LeFanu Lumsdaine. 2016. A Mechanization of the Blakers-Massey Connectivity Theorem in Homotopy Type Theory. In *31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16*. ACM, 565–574.

[14] Simon Huber. 2016. Canonicity for Cubical Type Theory. (2016). Preprint arXiv:1607.04156.

[15] Chris Kapulkin and Peter LeFanu Lumsdaine. 2012. The Simplicial Model of Univalent Foundations (after Voevodsky). (Nov. 2012). Preprint arXiv:1211.2851v4.

[16] Nicolai Kraus. 2016. Constructions with Non-Recursive Higher Inductive Types. In *31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS'16*. ACM, 595–604.

[17] Daniel R. Licata and Guillaume Brunerie. 2015. A Cubical Approach to Synthetic Homotopy Theory. In *30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS'15*. ACM, 92–103.

[18] Daniel R. Licata, Ian Orton, Andrew M. Pitts, and Bas Spitters. 2018. Internal Universes in Models of Homotopy Type Theory. (2018). Preprint arXiv:1801.07664.

[19] Peter LeFanu Lumsdaine and Michael Shulman. 2017. Semantics of higher inductive types. (May 2017). Preprint arXiv:1705.07088.

[20] Ian Orton and Andrew M. Pitts. 2016. Axioms for Modelling Cubical Type Theory in a Topos. In *25th EACSL Annual Conference on Computer Science Logic (CSL 2016) (LIPIcs)*, Vol. 62. 24:1–24:19.

[21] Emily Riehl and Michael Shulman. 2017. A type theory for synthetic ∞-categories. *Higher Structures* 1, 1 (2017), 147–224.

[22] Egbert Rijke. 2017. The join construction. (2017). Preprint arXiv:1701.07538v1.

[23] Christian Sattler. 2017. The Equivalence Extension Property and Model Structures. (2017). Preprint arXiv:1704.06911v1.

[24] Kristina Sojakova. 2016. The Equivalence of the Torus and the Product of Two Circles in Homotopy Type Theory. *ACM Transactions on Computational Logic* 17, 4 (Nov. 2016), 29:1–29:19.

[25] Andrew Swan. 2014. An Algebraic Weak Factorisation System on 01-Substitution Sets: A Constructive Proof. (2014). Preprint arXiv:1409.1829.

[26] The Univalent Foundations Program. 2013. *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study.

[27] Vladimir Voevodsky. 2014. The equivalence axiom and univalent models of type theory. (Talk at CMU on February 4, 2010). (2014). Preprint arXiv:1402.5556.

[28] Vladimir Voevodsky. 2015. An experimental library of formalized Mathematics based on the univalent foundations. *Mathematical Structures in Computer Science* 25 (2015), 1278–1294.