

Sequential Colimits in Homotopy Type Theory

Anonymous Author(s)

Abstract

Sequential colimits are an important class of higher inductive types. We present a self-contained proof of the conjecture that in homotopy type theory, sequential colimits appropriately commute with Σ -types. This result allows us to give short proofs of a number of useful corollaries, some of which were conjectured in other works: the commutativity of sequential colimits with identity types, with homotopy fibers, loop spaces, and truncations, and the preservation of the properties of truncatedness and connectedness under sequential colimits. Moreover, using Shulman’s recent interpretation of homotopy type theory into $(\infty, 1)$ -toposes, we are able to extend the classical analogue of our main result (by Hirschorn) from topological spaces to arbitrary $(\infty, 1)$ -toposes. All of our development is formalized in the proof assistant Lean.

Keywords sequential colimits, higher inductive types, homotopy type theory

1 Introduction

Homotopy type theory (HoTT) [The Univalent Foundations Program, Institute for Advanced Study 2013] is an extension of Martin-Löf’s dependent type theory [Martin-Löf 1975], in which types are thought of as spaces, dependent types as fibrations, terms of types as points of those spaces, and terms of the identity type $x = y$ for any $x, y : A$ as paths in A from x to y . This is the so-called homotopy interpretation of type theory, see [Awodey and Warren 2009]. Two further ingredients are then added to this type theory: the univalence axiom by Voevodsky [Kapulkin and Lumsdaine 2012; Voevodsky 2010, 2011], and higher inductive types by Bauer, Lumsdaine, Shulman, and Warren [Awodey et al. 2011; Lumsdaine 2011; Lumsdaine and Shulman 2017; Shulman 2011a]. The univalence axiom characterizes the identity type of the universe, and higher inductive types allow (among other things) constructions of new types by specifying their points and identifications between those points, in a fashion similar to introducing new algebras by generators and relations. Indeed, in [Sojakova 2016], Sojakova showed that higher inductive types are homotopy initial algebras. A semantics of higher inductive types suitable for interpreting the development in this paper is given in [Lumsdaine and Shulman 2017].

An important class of examples of higher inductive types are homotopy colimits, such as pushouts, coequalizers, and

sequential colimits. Sequential colimits in homotopy type theory were first studied by Brunerie [Brunerie 2016], who showed that the infinite dimensional sphere, defined as the colimit of the finite dimensional spheres, is contractible. A sequential colimit is the homotopy colimit of a diagram of the form

$$A(0) \xrightarrow{a(0)} A(1) \xrightarrow{a(1)} A(2) \xrightarrow{a(2)} \dots \quad (1)$$

Many interesting examples of types can be presented as sequential colimits. For example, van Doorn [van Doorn 2016] showed that the propositional truncation of a type can be obtained as a sequential colimit. In [Brunerie 2016], Brunerie performed the James construction in HoTT, showing that the loop space of a suspension can be presented as a sequential colimit. In [Rijke 2017], Rijke showed that n -truncations can be obtained as sequential colimits, via a construction that was generalized in [Christensen et al. 2018] to arbitrary reflective subuniverses. Moreover, stable homotopy groups and homology groups can be defined using a sequential colimit [Graham 2017]. We expect many more interesting applications to follow. For example, Shulman [Shulman 2013] suggested that the spectrification of a pre-spectrum can be presented as a sequential colimit, although a proof that the resulting pre-spectrum is indeed a spectrum is still work in progress.

In HoTT, the sequential colimit of a type sequence as in diagram 1 is defined as a higher inductive type called A_∞ . The point constructor of A_∞ is

$$\iota : \prod_{n:\mathbb{N}} A(n) \rightarrow A_\infty,$$

and the path constructor of A_∞ is

$$\kappa : \prod_{n:\mathbb{N}} \prod_{x:A(n)} \iota(n+1, a(n, x)) = \iota(n, x).$$

Just like all higher inductive types, the sequential colimit comes equipped with an induction principle and computation rules. Those principles essentially ensure that the type A_∞ indeed behaves as the homotopy colimit of the type sequence $A(0) \rightarrow A(1) \rightarrow \dots$. In fact, the induction principle implies a *dependent* universal property.

To explain our main technical result, we consider the situation where we are given a type family $B(n) : A(n) \rightarrow U$ over each $A(n)$, equipped with maps $b(n, a) : B(n, a) \rightarrow B(n+1, a(n, a))$ for each $a : A(n)$. We picture this situation as follows:

$$\begin{array}{ccccccc} B(0) & \xrightarrow{b(0)} & B(1) & \xrightarrow{b(1)} & B(2) & \xrightarrow{b(2)} & \dots \\ \downarrow & & \downarrow & & \downarrow & & \\ A(0) & \xrightarrow{a(0)} & A(1) & \xrightarrow{a(1)} & A(2) & \xrightarrow{a(2)} & \dots \end{array}$$

Here the double arrows indicate that the type family $B(n)$ is fibered over $A(n)$. Now we can consider the type sequence of total spaces (i.e. Σ -types)

$$\Sigma_{a:A(0)}B(0, a) \longrightarrow \Sigma_{a:A(1)}B(1, a) \longrightarrow \cdots,$$

and we write $\text{colim}(\Sigma(A, B))$ for its homotopy colimit. On the other hand, we can directly form a type family $B_\infty : A_\infty \rightarrow U$ and compare its total space with $\text{colim}(\Sigma(A, B))$. To construct B_∞ , we first form for each $a : A(n)$ the type sequence

$$B(n, a) \longrightarrow B(n+1, a(n, a)) \longrightarrow \cdots,$$

and we write $B_\infty(n, a)$ for the homotopy colimit of this sequence. Now we can construct an equivalence $B_\infty(n, a) \simeq B_\infty(n+1, a(n, a))$ for any $a : A(n)$, so by univalence we obtain a type family $B_\infty : A_\infty \rightarrow U$ as a result. Our main technical theorem, Theorem 5.1, states that we have a commuting triangle

$$\begin{array}{ccc} \text{colim}(\Sigma(A, B)) & \longrightarrow & \Sigma_{x:A_\infty} B_\infty(x) \\ & \searrow & \swarrow \\ & A_\infty & \end{array}$$

in which the top map is an equivalence. Among the chief consequences of our theorem (see section 7) is the characterization of the identity type of a sequential colimit, as a sequential colimit of identity types. This corollary can be seen as complementing the results of Kraus and von Raumer [Kraus and von Raumer 2019], who characterize the identity types of coequalizers and pushouts.

Related Work Our result has been conjectured for the setting of homotopy type theory by E. Rijke in 2015. The proof we give here, and the accompanying formalization, is the original one with which we settled the conjecture for homotopy type theory in 2017.

In the same year, R. Bocquet independently proved the analogous result in cubical type theory, using the proof assistant cubicaltt [Cohen et al. 2016]. His proof [Bocquet 2017] relies on the use of *indexed* higher inductive types, which we do not assume as part of our type theory. Furthermore, it is not known whether cubical type theory is conservative over the usual system of homotopy type theory as presented in [The Univalent Foundations Program, Institute for Advanced Study 2013]. In other words, it is not known whether an arbitrary statement of homotopy type theory that is provable in cubical type theory is also provable in homotopy type theory. To illustrate the importance of this point, we note that cubical type theory features many more judgmental equalities than standard homotopy type theory. There is a simple example that shows that adding judgmental equalities might change what is provable in the system: function extensionality follows from having an interval with judgmental computation rules at the point constructors [Shulman

2011b]. The question of whether cubical type theory is conservative over HoTT is an important open problem that was emphasized by C. Sattler.

In 2018, after our formalization had been completed, Bocquet gave an alternative proof of our theorem in Agda [Bocquet 2018] using Agda's implementation of coinductive types, which we likewise do not assume as part of our theory. Although homotopy coinductive types can be shown to exist in homotopy type theory [Ahrens et al. 2015], Bocquet made use of the native implementation of coinductive types in Agda <2.6 that was known to lead to an inconsistency [Agda Development Team 2019]. A more detailed discussion of these alternative proofs can be found in section 8.

On the semantics and a comparison to classical homotopy theory In [Shulman 2019] Shulman established that homotopy type theory is the internal language of an arbitrary $(\infty, 1)$ -topos. In other words, any $(\infty, 1)$ -topos is a model of Martin-Löf type theory with the all usual type constructors, and a univalent universe that is closed under those type constructors, with one notable exception: with the semantics of higher inductive types given by [Lumsdaine and Shulman 2017], the universe is not closed under higher inductive types. This condition is necessary, for example, if one wants to construct the n -spheres by recursion on the natural numbers, via the usual construction where the $(n+1)$ -sphere is the suspension of the n -sphere. The question whether universes can be closed under higher inductive types in the $(\infty, 1)$ -topos semantics of homotopy type theory remains one of the principal open problems.

In our development we make no use of the assumption that universes are closed under higher inductive types, hence our results are applicable to arbitrary $(\infty, 1)$ -toposes. We note that the Lean implementation of higher inductive types is such that universes are closed under higher inductive type, but for our proofs this fact is inessential.

The classical analogue of our main result can be found in [Hirschhorn 2019, Theorem 14.19]. The theorem is stated for diagrams of topological spaces, indexed by small filtered categories. The category (\mathbb{N}, \leq) that we implicitly use in our work is an example of a small filtered category (indeed, every ordinal gives an example).

Theorem 1.1 ([Hirschhorn 2019, Theorem 14.19]). *Let C be a small filtered category and let \mathcal{D} be a finite and acyclic category. If Y is a $C \times \mathcal{D}$ -diagram of topological spaces, then there is a natural zig-zag of weak equivalences showing*

$$\text{holim}_{\mathcal{D}} \text{hocolim}_C Y \simeq \text{hocolim}_C \text{holim}_{\mathcal{D}} Y.$$

This statement appears as the final theorem in a 75-page development, and it makes use of some techniques that we don't need in homotopy type theory. Most notably, the geometric realization is used throughout the article. There is no operation in homotopy type theory that corresponds to the geometric realization, because types do not have an

underlying set of points in the way topological spaces do. Since Hirschhorn's theorem is manifestly a statement about topological spaces, it does not directly apply to other $(\infty, 1)$ -toposes. Our result in homotopy type theory thus generalizes Hirschhorn's theorem to arbitrary $(\infty, 1)$ -toposes.

We furthermore point out that our proof requires very little prior setup and relies exclusively on suitable induction principles. Those are either given or proven along the way. In this sense, our work is entirely elementary. As a conceptual contribution, we present our proof in a fashion that aims to minimize the need for the explicit manipulation of paths – such as, *e.g.*, directly invoking the associativity of path concatenation – and relies instead on an appropriate generalization followed by a path induction. This approach has several advantages: it allows us to easily deal with certain typical equalities that are not definitional, we can carry out multiple simplifications at once, and, last but not least, the resulting proof is shorter and easier to follow on paper.

Overview In section 2 we briefly review the basics of homotopy type theory; in section 3 we establish some basic properties of type sequences and colimits; in section 4 we discuss fibered type sequences and construct the type family B_∞ ; in section 5 we state our main result and show that it follows from a more convenient induction principle on $\Sigma_{x:A_\infty} B_\infty(x)$; in section 6 we sketch the proof of the aforementioned induction principle; in section 7 we give some important corollaries of our main result; and in section 8 we say a few words about the formalization and the alternative proofs by Bocquet.

2 Homotopy Type Theory

We recall some basic notions and constructions from homotopy type theory; for a full account of the theory, see [The Univalent Foundations Program, Institute for Advanced Study 2013]. We distinguish two forms of equality in homotopy type theory: judgmental equality and typical equality. Judgmental equality is used for computation. When two expressions t_1 and t_2 for terms of type A describe the exact same term, we write $t_1 \equiv t_2$. For example, $1 + 1 \equiv 2$ of type \mathbb{N} . The second form of equality is typical equality, which is introduced in the form of Martin-Löf's identity type. For any two terms x and y of type A there is a type $x = y$, and proving that $x = y$ holds amounts to constructing a term of the type $x = y$. The constant path from x to itself is called refl_x . For any two paths $p : x = y$ and $q : y = z$, we write $p \cdot q$ for their concatenation. Concatenation is defined in such a way that $\text{refl}_x \cdot \text{refl}_x \equiv \text{refl}_x$, and there is an identification $(p \cdot q) \cdot r \equiv p \cdot (q \cdot r)$ for any three concatenatable paths. Moreover, any function $f : A \rightarrow B$ respects paths: for any path $p : x = y$ in A there is a path $\text{ap}_f(p) : f(x) = f(y)$.

We recall that a homotopy between two (dependent) maps $f, g : \Pi_{x:A} B(x)$ is a term of type $f \sim g := \Pi_{x:A} f(x) = g(x)$. An equivalence is then defined as a map $f : A \rightarrow B$ equipped

with a right and a left inverse, *i.e.* maps $g : B \rightarrow A$ and $h : B \rightarrow A$ with homotopies $f \circ g \sim \text{id}_B$ and $h \circ f \sim \text{id}_A$. We write $A \simeq B$ for the type of equivalences from A to B . The univalence axiom asserts that the canonical map $(A = B) \rightarrow (A \simeq B)$ is an equivalence. We will write $\bar{p} : A \rightarrow B$ for the function associated to a path $p : A = B$ in \mathcal{U} . The function extensionality principle (which follows from univalence) asserts that for any $f, g : \Pi_{x:A} B(x)$, the map $(f = g) \rightarrow (f \sim g)$ is an equivalence. For $f : A \rightarrow B$ and $y : B$, the homotopy fiber of f at y is defined as $\text{fib}_f(y) := \Sigma_{x:A} f(x) = y$.

An important concept in homotopy type theory is that of transport. Given a type family $B : A \rightarrow \mathcal{U}$ and a path $p : x = y$ in A , we have a function $p \#_B : B(x) \rightarrow B(y)$, with $\text{refl}_x \#_B \equiv \text{id}$. Using transport, we can also show that dependent functions respect paths: for any dependent function $f : \Pi_{x:A} B(x)$ and any path $p : x = y$ in A , we can't directly compare $f(x)$ of type $B(x)$ with $f(y)$ of type $B(y)$. Still, there is a path $\text{apd}_f(p) : p \#_B f(x) = f(y)$ in $B(y)$.

The notion that a type A is n -truncated is defined by recursion on $n : \mathbb{Z}$, $n \geq -2$. We say that A is (-2) -truncated or contractible if we can find an inhabitant of $\Sigma_{x:A} \Pi_{y:A} x = y$. The type A is $(n + 1)$ -truncated if for all $x, y : A$ the type $x = y$ is n -truncated. For any type A we write $\|A\|_n$ for its n -truncation, *i.e.* $\|A\|_n$ is an n -truncated type equipped with a map $|-|_n : A \rightarrow \|A\|_n$ such that for any n -truncated type B the precomposition map $(\|A\|_n \rightarrow B) \rightarrow (A \rightarrow B)$ is an equivalence. A type A is n -connected if $\|A\|_n$ is contractible. A function $f : A \rightarrow B$ is called n -truncated (n -connected) if for all $y : B$ the type $\text{fib}_f(y)$ is n -truncated (n -connected). Properties of these notions are established in Chapter 7 of [The Univalent Foundations Program, Institute for Advanced Study 2013].

3 Type Sequences and Sequential Colimits

Our main ingredient, the sequential colimit, is a colimit of a diagram of a particular shape:

Definition 3.1. A *type sequence* is a pair $(A, a) : \text{Seq}$ with

- $A : \mathbb{N} \rightarrow \mathcal{U}$
- $a : \Pi_{n:\mathbb{N}} A(n) \rightarrow A(n + 1)$

Morphisms of type sequences are natural transformations:

Definition 3.2. Let (A, a) and (A', a') be type sequences. A *natural transformation* $(A, a) \rightarrow (A', a')$ is a pair (τ, H) consisting of a family of maps

$$\tau : \Pi_{n:\mathbb{N}} A(n) \rightarrow A'(n)$$

and a family of homotopies $H(n)$ witnessing that the diagram

$$\begin{array}{ccccccc} A(0) & \xrightarrow{a(0)} & A(1) & \xrightarrow{a(1)} & A(2) & \xrightarrow{a(2)} & \dots \\ \downarrow \tau(0) & & \downarrow \tau(1) & & \downarrow \tau(2) & & \\ A'(0) & \xrightarrow{a'(0)} & A'(1) & \xrightarrow{a'(1)} & A'(2) & \xrightarrow{a'(2)} & \dots \end{array}$$

commutes, *i.e.* $H_n : \tau(n + 1) \circ a(n) \sim a'(n) \circ \tau(n)$.

A natural equivalence is a natural transformation (τ, H) where $\tau(n)$ is an equivalence for all n .

Definition 3.3. A colimit $\text{colim}(A, a)$ of a type sequence (A, a) is the higher inductive type generated by the following constructors:

$$\begin{aligned} \iota &: \prod_{n:\mathbb{N}} A(n) \rightarrow \text{colim}(A, a) \\ \kappa &: \prod_{n:\mathbb{N}} \prod_{a:A(n)} \iota(n+1, a(n, a)) = \iota(n, a) \end{aligned}$$

The constructor ι gives the canonical injection of $a : A(n)$ into the colimit, whereas the constructor κ provides the gluing that ensures that the respective injections of the points $a : A(n)$ and $a(n, a) : A(n+1)$ into the colimit coincide, which in the HoTT world means that they are typically equal. The higher dimensionality of this inductively-defined type comes from the fact that κ does not construct a term of $\text{colim}(A, a)$ itself but rather of the identity type over $\text{colim}(A, a)$. We will sometimes denote $\text{colim}(A, a)$ by $\text{colim}_n(A(n))$ or A_∞ if the maps can be inferred from the context.

The constructors in Definition 3.3 are accompanied by the following principle of induction: given

- $E : \text{colim}(A, a) \rightarrow U$
- $e : \prod_{n:\mathbb{N}} \prod_{a:A(n)} E(\iota(n, a))$
- $p : \prod_{n:\mathbb{N}} \prod_{a:A(n)} \kappa(n, a) \#_E e(n+1, a(n, a)) = e(n, a)$

we have a map $\text{ind}(E, e, p) : \prod_{x:\text{colim}(A, a)} E(x)$ such that for any $n : \mathbb{N}, a : A(n)$,

- $\text{ind}(E, e, p) \iota(n, a) \equiv e(n, a)$
- $\text{ap}_{\text{ind}(E, e, p)}(\kappa(n, a)) = p(n, a)$

The induction principle implies the following recursion principle: given

- $E : U$
- $e : \prod_{n:\mathbb{N}} A(n) \rightarrow E$
- $p : \prod_{n:\mathbb{N}} \prod_{a:A(n)} e(n+1, a(n, a)) = e(n, a)$

there is a function $\text{rec}(E, e, p) : \text{colim}(A, a) \rightarrow E$ such that for any $n : \mathbb{N}, a : A(n)$,

- $\text{rec}(E, e, p) \iota(n, a) \equiv e(n, a)$
- $\text{ap}_{\text{rec}(E, e, p)}(\kappa(n, a)) = p(n, a)$

One important consequence of the induction principle is:

Lemma 3.4 (Uniqueness property of the sequential colimit). *Two functions $F_1, F_2 : \text{colim}(A, a) \rightarrow E$ out of the sequential colimit are equal iff there is a homotopy*

- $\alpha : \prod_{n:\mathbb{N}} \prod_{a:A(n)} F_1(\iota(n, a)) = F_2(\iota(n, a))$

such that the diagram of identifications below commutes for all $n : \mathbb{N}, a : A(n)$.

$$\begin{array}{ccc} F_1(\iota(n+1, a(n, a))) & \xrightarrow{\text{ap}_{F_1}(\kappa(n, a))} & F_1(\iota(n, a)) \\ \alpha(n+1, a(n, a)) \downarrow & & \downarrow \alpha(n, a) \\ F_2(\iota(n+1, a(n, a))) & \xrightarrow{\text{ap}_{F_2}(\kappa(n, a))} & F_2(\iota(n, a)) \end{array}$$

We can use this uniqueness property to establish that the colimit $\text{colim}(A, a)$ of the original sequence is in fact equal to the colimit of the sequence

$$A(1) \xrightarrow{a(1)} A(2) \xrightarrow{a(2)} A(3) \xrightarrow{a(3)} \dots$$

which drops the first index. This is intuitively clear since any point $\iota(0, a)$ corresponding to $a : A(0)$ is already present in the colimit, up to typical equality, as the point $\iota(1, a(0, a))$ corresponding to $a(0, a) : A(1)$. Formally:

Lemma 3.5. *We have*

$$\text{colim}(n \mapsto A(n+1), n \mapsto a(n+1)) = \text{colim}(A, a)$$

as witnessed by the equivalence

$$\begin{aligned} \iota(n, a) &\mapsto \iota(n+1, a) \\ \kappa(n, a) &\mapsto \kappa(n+1, a) \end{aligned}$$

Proof. We define the map in the opposite direction by

$$\begin{aligned} \iota(n, a) &\mapsto \iota(n, a(n, a)) \\ \kappa(n, a) &\mapsto \kappa(n, a(n, a)) \end{aligned}$$

Showing that these maps are indeed inverse is now an easy consequence of the uniqueness property of the sequential colimit. \square

A term $a : A(n)$ can be *lifted* to the term $a(n, a) : A(n+1)$, which can then be lifted to the term $a(n+1, a(n, a)) : A(n+2)$, and so on. Formally, let $\Sigma A := \sum_{n:\mathbb{N}} A(n)$ and define the lifting $(\cdot)^+ : \Sigma A \rightarrow \Sigma A$ by $(n, a)^+ := (n+1, a(n, a))$. We define the k -fold lifting $(\cdot)^{+k} : \Sigma A \rightarrow \Sigma A$ inductively by $x^{+0} := x$ and $x^{+(k+1)} := (x^{+k})^+$. Lifting is associative up to typical equality: we define $q(n, a, k) : (n, a)^{+k} = (n, a)^{+k+1}$ inductively by $q(n, a, 0) := \text{refl}((n, a)^+)$ and $q(n, a, k+1) := \text{ap}_{(\cdot)^+}(q(n, a, k))$.

The uniqueness property of the colimit thus allows us to characterize the type of maps out of the sequential colimit $\text{colim}(A, a)$ as follows:

Lemma 3.6. *We have*

$$(\text{colim}(A, a) \rightarrow E) = \left(\sum_{e:\Sigma A \rightarrow E} \prod_{x:\Sigma A} e(x^+) = e(x) \right)$$

A natural transformation between type sequences induces a map between the respective colimits:

Lemma 3.7. *Let $(\tau, H) : (A, a) \rightarrow (A', a')$ be a natural transformation. Then:*

1. *We get a function $\text{colim}(\tau, H)$ or $\tau_\infty : A_\infty \rightarrow A'_\infty$.*
2. *The sequential colimit is 1-functorial. This means the following three things: If $(\sigma, K) : (A', a') \rightarrow (A'', a'')$ then $(\sigma \circ \tau)_\infty \sim \sigma_\infty \circ \tau_\infty$. Moreover, $1_\infty \sim \text{id}$, where 1 is the identity natural transformation. Lastly, if $(\tau', H') : (A, a) \rightarrow (A', a')$ and $q : \prod_{n:\mathbb{N}} \tau(n) \sim \tau'(n)$ and we can fill the following square for all $a : A(n)$*

$$\begin{array}{c} \tau(n+1, a(n, a)) \xrightarrow{\text{ap}_{a'(n)}(q(n, a))} \tau'(n+1, a(n, a)) \\ \parallel^{H(n, a)} \qquad \qquad \qquad \parallel^{H'(n, a)} \\ a'(n, \tau(n, a)) \xrightarrow{\text{ap}_{a'(n)}(q(n, a))} a'(n, \tau'(n, a)) \end{array}$$

then $\tau_\infty \sim \tau'_\infty$.

3. If τ is a natural equivalence then τ_∞ is an equivalence.

Proof.

1. We define $\tau_\infty(u(n, a)) := u(n, \tau(n, a))$ and

$$\begin{aligned} \text{ap}_{\tau_\infty}(\kappa(n, a)) &:= \text{ap}_{u(n+1)}(H(a)) \cdot \kappa(n, \tau(n, a)) \\ &: u(n+1, \tau(n+1, a(n, a))) = u(n, \tau(n, a)). \end{aligned}$$

2. All three parts are by straightforward induction on A_∞ .

3. We define $(\tau_\infty)^{-1} := (\tau^{-1})_\infty$ where τ^{-1} is the natural transformation by inverting τ_n for each n . Now we can check that this is really the inverse by using all three parts of the 1-functoriality:

$$\tau_\infty^{-1} \circ \tau_\infty \sim (\tau^{-1} \circ \tau)_\infty \sim 1_\infty \sim \text{id}_{A_\infty}$$

□

4 Fibered Type Sequences

We now shift our attention to a dependent version of type sequences:

Definition 4.1. A type sequence *fibered over* (A, a) is a pair $(B, b) : \text{FibSeq}(A, a)$ with

- $B : \Sigma A \rightarrow \mathcal{U}$
- $b : \Pi_{x:\Sigma A} B(x) \rightarrow B(x^+)$

which can be visualized as the family of type sequences below, one for each $x : \Sigma A$.

$$B(x) \xrightarrow{b(x)} B(x^+) \xrightarrow{b(x^+)} B(x^{++}) \xrightarrow{b(x^{++})} \dots$$

We say that (B, b) is *equifibered* if each $b(x)$ is an equivalence.

Lemma 3.6 plus univalence imply that the type of fibrations over $\text{colim}(A, a)$ is equal to the type of equifibered sequences over (A, a) :

Lemma 4.2. *The type $\text{colim}(A, a) \rightarrow \mathcal{U}$ corresponds exactly to equifibered type sequences over (A, a) , i.e.*

$$\begin{aligned} &\text{colim}(A, a) \rightarrow \mathcal{U} \\ &= \Sigma_{C:\Pi_{n:\mathbb{N}}A(n) \rightarrow \mathcal{U}} \Pi_{n:\mathbb{N}} \Pi_{a:A(n)} C(n+1, a(n, a)) \\ &\simeq C(n, a) \end{aligned}$$

Since for any $n : \mathbb{N}$ we can form the type $\Sigma_{a:A(n)} B(n, a)$, there is a natural way to combine (A, a) and (B, b) into a type sequence:

Definition 4.3. We define the type sequence $\Sigma_{(A, a)}(B, b)$ as the pair

- $n \mapsto \Sigma_{a:A(n)} B(n, a)$
- $n \mapsto ((a, b) \mapsto (a(n, a), b((n, a), b)))$

We have the canonical projection from $\text{colim}(\Sigma_{(A, a)}(B, b))$ to $\text{colim}(A, a)$ defined by:

$$\begin{aligned} u(n, (a, b)) &\mapsto u(n, a) \\ \kappa(n, (a, b)) &\mapsto \kappa(n, a) \end{aligned}$$

We can informally describe the type $\text{colim}(\Sigma_{(A, a)}(B, b))$ as a “colim after Σ ”. The natural questions now are, what does the dual type “ Σ after colim” look like, and are the two types equal? We will answer the former in this section and the latter in the rest of the paper.

A dependent sum of colimits has the form $\Sigma_{x:\text{colim}(A, a)} ??(x)$, where “??” is obtained from the fibered typed sequence (B, b) by using the colimit operation in some way. So let us denote this type family by B_∞ .

We define $B_\infty : A_\infty \rightarrow \mathcal{U}$ by giving $C_\infty : \Pi_{n:\mathbb{N}} A(n) \rightarrow \mathcal{U}$ and $c : \Pi_{n:\mathbb{N}} \Pi_{a:A(n)} C_\infty(n+1, a(n, a)) \simeq C_\infty(n, a)$. Since (B, b) defines a type sequence for every pair $(n, a) : \Sigma A$, the main candidate for $C_\infty(n, a)$ is its colimit:

$$C_\infty(n, a) := \text{colim}(k \mapsto B((n, a)^{++k}), k \mapsto b((n, a)^{++k}))$$

To define $c(x)$, we want to appeal to Lemma 3.5 to obtain a composition of equivalences as follows:

$$\begin{aligned} &C_\infty(n+1, a(n, a)) \\ &\equiv \text{colim}(k \mapsto B((n, a)^{++k}), k \mapsto b((n, a)^{++k})) \\ &\simeq \text{colim}(k \mapsto B((n, a)^{++k+1}), k \mapsto b((n, a)^{++k+1})) \\ &\simeq \text{colim}(k \mapsto B((n, a)^{++k}), k \mapsto b((n, a)^{++k})) \\ &\equiv C_\infty(n, a) \end{aligned}$$

The second equivalence, which we call $e_2(n, a)$, comes from Lemma 3.5. To construct the first equivalence, which we call $e_1(n, a)$, we use the functoriality of colim, Lemma 3.7. The natural transformation is given by

$$Q(n, a, k) := \overline{\text{ap}_B(q(n, a, k))}$$

Recall that we use the overline for the function transporting along a path between types. For the naturality square, we need a filler of the following diagram witnessed by a homotopy $\delta(n, a, k)$.

$$\begin{array}{ccc} B((n, a)^{++k}) & \xrightarrow{Q(n, a, k)} & B((n, a)^{++k+1}) \\ b((n, a)^{++k}) \downarrow & \uparrow \delta(n, a, k) & \downarrow b((n, a)^{++k+1}) \\ B((n, a)^{++k+1}) & \xrightarrow{Q(n, a, k+1)} & B((n, a)^{++k+2}) \end{array}$$

To construct this homotopy, we expand the definition of Q :

$$\begin{array}{ccc} B((n, a)^{++k}) & \xrightarrow{\overline{\text{ap}_B(q(n, a, k))}} & B((n, a)^{++k+1}) \\ b((n, a)^{++k}) \downarrow & & \downarrow b((n, a)^{++k+1}) \\ B((n, a)^{++k+1}) & \xrightarrow{\overline{\text{ap}_B(\text{ap}_{(\cdot)^+}(q(n, a, k)))}} & B((n, a)^{++k+2}) \end{array}$$

We can fill this diagram by generalizing the path $q(n, a, k) : (n, a)^{++k} = (n, a)^{+k+1}$ together with its endpoints and doing path induction. This gives the map $e_1(n, a)$. Moreover, (Q, δ) is a natural equivalence, hence $e_1(n, a)$ is an equivalence by Lemma 3.7.3.

Finally, the composition $e_2(n, a) \circ e_1(n, a)$ gives us the desired equivalence $c(n, a) : C_\infty(n+1, a(n, a)) \rightarrow C_\infty(n, a)$. Thus, $c(n, a)$ is such that we have:

- $c(n, a) \iota(k, b) \equiv \iota(k+1, Q(n, a, k, b))$
- $\beta_c(n, a) : \text{ap}_{c(n, a)}(\kappa(k, b)) = \text{ap}_{\iota(k+2)}(\delta(n, a, k, b)) \cdot \kappa(k+1, Q(n, a, k, b))$

On numerals k , we have $c(n, a) \iota(k, b) \equiv \iota(k+1, b)$ and $\text{ap}_{c(n, a)}(\kappa(k, b)) = \kappa(k+1, b)$.

We have thus constructed $B_\infty : A_\infty \rightarrow \mathcal{U}$ such that we have:

- $B_\infty(\iota(n, a)) \equiv C_\infty(n, a)$
- $c_\star(n, a) : \text{ap}_{B_\infty}(\kappa(n, a)) = c(n, a)$

5 Colimits and Sums

To show $\text{colim}(\Sigma_{(A, a)}(B, b)) \simeq \Sigma_{x:A_\infty} B_\infty(x)$, we want to construct mutually inverse functions between these types. From left to right, the only reasonable action on points is to map $\iota(n, (a, b))$ to the pair $(\iota(n, a), \iota(0, b))$, which makes sense since $b : B(n, a)$. Mapping the path constructor $\kappa(n, (a, b)) : \iota(n+1, (a(n, a), b((n, a), b))) = \iota(n, (a, b))$ requires a path

$$\begin{array}{c} (\iota(n+1, a(n, a)), \iota(0, b((n, a), b))) \\ \downarrow ?? \\ (\iota(n, a), \iota(0, b)) \end{array}$$

The constructor $\kappa(0, b)$ for the sequential colimit $C_\infty(n, a)$ gives us a path $\iota(1, b((n, a), b)) = \iota(0, b)$ that we can use as follows:

$$\begin{array}{c} (\iota(n+1, a(n, a)), \iota(0, b((n, a), b))) \\ \downarrow ?? \\ (\iota(n, a), \iota(1, b((n, a), b))) \\ \downarrow \text{ap}_{\iota(n, a)}(\kappa(0, b)) \\ (\iota(n, a), \iota(0, b)) \end{array}$$

In the remaining path, the constructor $\kappa(n, a)$ for the sequential colimit A_∞ gives us equality of the first components, and hence a map $\text{ap}_{B_\infty}(\kappa(n, a)) : C_\infty(n+1, a(n, a)) \rightarrow C_\infty(n, a)$. This map is equal to $c(n, a)$, as witnessed by $c_\star(n, a)$, and the latter carries $\iota(0, b((n, a), b))$ precisely to $\iota(1, b((n, a), b))$. We can generalize this situation as follows: given

- $x_1, x_2 : A_\infty$ and $\alpha : x_1 = x_2$
- $F : B_\infty(x_1) \rightarrow B_\infty(x_2)$
- $F_\star : \text{ap}_{B_\infty}(\alpha) = F$
- $y : B_\infty(x_1)$

we have a path $\Delta(\alpha, F_\star, y) : (x_1, y) = (x_2, F(y))$ defined by induction on α and F_\star . This completes the definition of a function from left to right, and allows us to concisely state our main result:

Theorem 5.1 (Interaction between colim and Σ). *We have:*

$$\text{colim}(\Sigma_{(A, a)}(B, b)) = \Sigma_{x:A_\infty} B_\infty(x)$$

as witnessed by the equivalence F below:

$$\iota(n, (a, b)) \mapsto (\iota(n, a), \iota(0, b))$$

$$\kappa(n, (a, b)) \mapsto \Delta\left(\kappa(n, a), c_\star(n, a), \iota(0, b((n, a), b))\right) \cdot \text{ap}_{(\iota(n, a), -)}(\kappa(0, b))$$

that commutes with the canonical projections:

$$\begin{array}{ccc} \text{colim}(\Sigma_{(A, a)}(B, b)) & \xrightarrow{F} & \Sigma_{x:A_\infty} B_\infty(x) \\ & \searrow & \swarrow \pi_1 \\ & \text{colim}(A, a) \equiv A_\infty & \end{array}$$

with the one on the left as given in Definition 4.3.

Proof of commutativity. We use the uniqueness property of the colimit. On points $\iota(n, (a, b))$ the maps agree definitionally. To show that $\kappa(n, a) = \text{ap}_{\pi_1}(\text{ap}_F(\kappa(n, (a, b))))$, we use the following easy generalization: given

- $x_1, x_2 : A_\infty$ and $\alpha : x_1 = x_2$
- $y_1 : B_\infty(x_1)$ and $y_2 : B_\infty(x_2)$
- $F : B_\infty(x_1) \rightarrow B_\infty(x_2)$
- $F_\star : \text{ap}_{B_\infty}(\alpha) = F$
- $\beta : F(y_1) = y_2$

we have $\alpha = \text{ap}_{\pi_1}(\Delta(\alpha, F_\star, y_1) \cdot \text{ap}_{(x_2, -)}(\beta))$. \square

Now it “only” remains to show that F is indeed an equivalence, by exhibiting an inverse. However, induction on the sum of sequential colimits is significantly harder than induction on the sequential colimit of the sum: the former requires two nested colimit inductions and results not in a *point* case and a *path* case, as before, but rather in *point-point*, *point-path*, *path-point*, *path-path* cases.

Worse yet, it appears we may have to carry out this form of induction *twice*: once when constructing the inverse to F and again when proving that the two maps compose to the identity on $\Sigma_{x:A_\infty} B_\infty(x)$. Or do we? If the sum of colimits is to be equal to the colimit of the sum, then the two types better have the same induction principle! In other words, we should be able to construct a map out of $\Sigma_{x:A_\infty} B_\infty(x)$ as if we were constructing a map out of $\text{colim}(\Sigma_{(A, a)}(B, b))$, using F to appropriately mediate between the two types:

Lemma 5.2 (Induction on the sum of sequential colimits). *Given*

- $E : (\Sigma_{x:A_\infty} B_\infty(x)) \rightarrow \mathcal{U}$
- $e : \prod_{n:\mathbb{N}} \prod_{a:A(n)} \prod_{b:B(n, a)} E(F(\iota(n, (a, b))))$

- $p : \prod_{n:\mathbb{N}} \prod_{a:A(n)} \prod_{b:B(n,a)} \text{ap}_F(\kappa(n, (a, b))) \#_E e(n+1, a(n, a), b((n, a), b)) = e(n, a, b)$

there is a function $G : \prod_{y:(\sum_{x:A_\infty} B_\infty(x))} E(y)$ such that for any $n : \mathbb{N}, a : A(n), b : B(n, a)$,

- $G(F(\iota(n, (a, b)))) \equiv e(n, a, b)$
- $\text{apd}_G(\text{ap}_F(\kappa(n, (a, b)))) = p(n, a, b)$

Of course, to establish this lemma we still need to carry out the more difficult nested induction – but only once, and then our work is essentially done. Lemma 5.2 easily implies the corresponding recursion principle:

Lemma 5.3 (Recursion on the sum of sequential colimits). *Given*

- $E : \mathcal{U}$
- $e : \prod_{n:\mathbb{N}} \prod_{a:A(n)} B(n, a) \rightarrow E$
- $p : \prod_{n:\mathbb{N}} \prod_{a:A(n)} \prod_{b:B(n,a)} e(n+1, a(n, a), b((n, a), b)) = e(n, a, b)$

there is a function $G : (\sum_{x:A_\infty} B_\infty(x)) \rightarrow E$ such that for any $n : \mathbb{N}, a : A(n), b : B(n, a)$,

- $G(F(\iota(n, (a, b)))) \equiv e(n, a, b)$
- $\text{ap}_G(\text{ap}_F(\kappa(n, (a, b)))) = p(n, a, b)$

Lemma 5.2 and the uniqueness property of the sequential colimit now easily imply the following:

Lemma 5.4 (Uniqueness property of the sum of sequential colimits). *Two maps $G_1, G_2 : (\sum_{x:A_\infty} B_\infty(x)) \rightarrow E$ out of the sum of colimits are equal iff the maps $G_1 \circ F$ and $G_2 \circ F$ are, i.e. F is epic.*

The proof of our main result, Theorem 5.1, is now easy:

Proof of equivalence. We construct the inverse F^{-1} by recursion on $\sum_{x:A_\infty} B_\infty(x)$ as in Lemma 5.3:

$$\begin{aligned} F(\iota(n, (a, b))) &\mapsto \iota(n, (a, b)) \\ \text{ap}_F(\kappa(n, (a, b))) &\mapsto \kappa(n, (a, b)) \end{aligned}$$

The equality $F^{-1} \circ F = \text{id}$ is now immediate from the uniqueness property of the sequential colimit and the equality $F \circ F^{-1} = \text{id}$ from the uniqueness property of the sum of sequential colimits. \square

In the next section we sketch the proof of Lemma 5.2.

6 Induction on the Sum of Sequential Colimits

To prove Lemma 5.2, we want to construct the curried version of G , the map $x, y \mapsto G(x, y) : \prod_{x:A_\infty} \prod_{y:B_\infty(x)} E(x, y)$, by nested sequential colimit induction. For the outer induction on A_∞ , we need to map $\iota(n, a)$ to a function

- $g(n, a) : \prod_{y:C_\infty(n,a)} E(\iota(n, a), y)$

and $\kappa(n, a)$ to an equality

$$\kappa(n, a) \#_{(x \mapsto \prod_{y:B_\infty(x)} E(x, y))} g(n+1, a(n, a)) = g(n, a) \quad (1)$$

We want to define $g(n, a)$ by induction on the sequential colimit $C_\infty(n, a)$. We thus need h and H such that for any $k : \mathbb{N}$ and $b : B((n, a)^{+k})$,

- $h(n, a, k, b) : E(\iota(n, a), \iota(k, b))$
- $H(n, a, k, b) : \kappa(k, b) \#_{E(\iota(n, a), -)} h(n, a, k+1, b((n, a)^{+k}, b)) = h(n, a, k, b)$

The obvious way to proceed now would be to begin defining h and H right away and establish equality (1) post-hoc. This may result in more work than necessary though: perhaps if we examine this equality first we can get some insight into how to best define h and H so that only a (relatively) small amount of effort is needed for the *path-point* and *path-path* cases.

We start by reformulating equality 1 in a way that makes it easier to understand. The maps $g(n+1, a(n, a))$ and $g(n, a)$ have different domains: the former takes arguments from the sequential colimit $C_\infty(n+1, a(n, a))$ and the latter from $C_\infty(n, a)$. But these sequential colimits are equivalent via $c(n, a)$, so relating the maps $g(n+1, a(n, a))$ and $g(n, a)$ comes down to relating $g(n+1, a(n, a), b)$ and $g(n, a, c(n, a), b)$ for any argument $b : C_\infty(n+1, a(n, a))$. The respective values lie in different fibers of E , namely over $(\iota(n+1, a(n, a)), b)$ versus over $(\iota(n, a), c(n, a), b)$. But this is fine: the path

$$\Delta(\kappa(n, a), c_\star(n, a), b)$$

equates the pairs.

Thus, we have the following more verbose but easier to use reformulation:

$$\begin{aligned} \prod_{y:C_\infty(n+1, a(n, a))} g(n, a, c(n, a), y) = \\ \Delta(\kappa(n, a), c_\star(n, a), y) \#_E g(n+1, a(n, a), y) \quad (2) \end{aligned}$$

Of course, we must show this is equivalent to the original formulation. We do so via a generalization: given

- $x_1, x_2 : A_\infty$ and $\alpha : x_1 = x_2$
- $F : B_\infty(x_1) \rightarrow B_\infty(x_2)$
- $F_\star : \text{ap}_{B_\infty}(\alpha) = F$
- $f_1 : \prod_{y:B_\infty(x_1)} E(x_1, y)$
- $f_2 : \prod_{y:B_\infty(x_2)} E(x_2, y)$

we have an equivalence $l(\alpha, F_\star, f_1, f_2) :$

$$\alpha \#_{(x \mapsto \prod_{y:B_\infty(x)} E(x, y))} f_1 = f_2 \simeq$$

$$\prod_{y:B_\infty(x_1)} f_2(F(y)) = \Delta(\alpha, F_\star, y) \#_E f_1(y)$$

We define l by induction on α and F_\star . This reduces the above equivalence to $(f_1 = f_2) \simeq (f_1 \sim f_2)$, witnessed by the map $\text{hap}(f_1, f_2)$.

It thus suffices to produce a witness $\omega(n, a)$ for (2), which will give us the desired witness for (1), explicitly described

below:

$$\left(\kappa(n, a), c_{\star}(n, a), g(n+1, a(n, a)), g(n, a) \right)^{-1} \omega(n, a)$$

We construct $\omega(n, a)$ by induction on the sequential colimit $C_{\infty}(n+1, a(n, a))$. So for any $k : \mathbb{N}$ and $b : B((n, a)^{++k})$ we need

$$\bullet \mu(n, a, k, b) : g(n, a, c(n, a) \iota(k, b)) = \Delta(\kappa(n, a), c_{\star}(n, a), \iota(k, b)) \#_E g(n+1, a(n, a), \iota(k, b))$$

and an equality:

$$\begin{aligned} \kappa(k, b) \#_{y \mapsto g(n, a, c(n, a) y) = \Delta(\kappa(n, a), c_{\star}(n, a), y) \#_E g(n+1, a(n, a), y)} \\ \mu(n, a, k+1, b((n, a)^{++k}) b) = \mu(n, a, k, b) \end{aligned} \quad (3)$$

We can now divide the proof into the following steps:

- The *point-point* case: defining h .
- The *path-point* case: defining μ .
- The *point-path* case: defining H .
- The *path-path* case: showing that 3 holds.
- Establishing the computation rules.

6.1 The *point-point* and *path-point* cases

We first simplify the type of μ . By definition of g we have

$$g(n+1, a(n, a), \iota(k, b)) \equiv h(n+1, a(n, a), k, b)$$

and similarly

$$\begin{aligned} g(n, a, c(n, a) \iota(k, b)) &\equiv g(n, a, \iota(k+1, Q(n, a, k) b)) \\ &\equiv h(n, a, k+1, Q(n, a, k) b) \end{aligned}$$

hence we need

$$\bullet \mu : h(n, a, k+1, Q(n, a, k) b) = \Delta(\kappa(n, a), c_{\star}(n, a), \iota(k, b)) \#_E h(n+1, a(n, a), k, b)$$

We recall that the map $Q(n, a, k) := \overline{\text{ap}_B(q(n, a, k))}$ is an equivalence. So the above equality shows how to express $h(\cdot, \cdot, k+1, \cdot)$ in terms of $h(\cdot, \cdot, k, \cdot)$, giving us an inductive definition of h . We thus aim to define a function $k, n, a, b \mapsto h(n, a, k, b) : \prod_{k:\mathbb{N}} \prod_{n:\mathbb{N}} \prod_{a:A(n)} \prod_{b:B((n, a)^{++k})} E(\iota(n, a), \iota(k, b))$ by induction on k . In the zero case, we need a point in the fiber $E(\iota(n, a), \iota(0, b))$ and this is just the assumption $e(n, a, b) : E(F(\iota(n, a, b)))$. In the successor case, we would like to define $h(n, a, k+1, b)$ in terms of $h(\cdot, \cdot, k, \cdot)$ as in the type of μ , but we cannot do so directly because b does not have the form $b := Q(n, a, k) \dots$ *on the nose*, only up to typal equality. Working with this typal equality is not very convenient either because the type of $h(n, a, k+1, b)$ itself depends on b so we would additionally have to factor in a transport. Instead, we use the following trivial generalization: given

- $A, B : \mathcal{U}$ and $\alpha : A = B$
- $P : B \rightarrow \mathcal{U}$
- $f : \prod_{a:A} P(\overline{\alpha}(a))$

we have a map $J(\alpha, P, f) : \prod_{b:B} P(b)$ with the computation rule

$$\beta_J(\alpha, P, f) : \prod_{a:A} J(\alpha, P, f, \overline{\alpha}(a)) = f(a)$$

In other words, when defining a function out of B we can instead assume the arguments come from A . Giving a name to this not-particularly-deep observation enables us to concisely define h and μ :

$$h(n, a, 0, b) := e(n, a, b)$$

$$h(n, a, k+1) := J(\text{ap}_B(q(n, a, k)),$$

$$b : B((n, a)^{++k+1}) \mapsto E(\iota(n, a), \iota(k+1, b)),$$

$$b : B((n, a)^{++k}) \mapsto$$

$$\Delta(\kappa(n, a), c_{\star}(n, a), \iota(k, b)) \#_E h(n+1, a(n, a), k, b))$$

$$\mu(n, a, k) := \beta_J(\text{ap}_B(q(n, a, k)),$$

$$b : B((n, a)^{++k+1}) \mapsto E(\iota(n, a), \iota(k+1, b)),$$

$$b : B((n, a)^{++k}) \mapsto$$

$$\Delta(\kappa(n, a), c_{\star}(n, a), \iota(k, b)) \#_E h(n+1, a(n, a), k, b))$$

6.2 The *point-path* and *path-path* cases

As in the previous subsection, we try to show that 3 holds before defining H , gaining an insight into what H should look like. We again start by replacing 3 with something equivalent but easier to understand. To figure out what the transport does, we would like to appeal to path induction. So the first step is to replace the path $\kappa(k, b)$ together with its endpoints $\iota(k+1, b((n, a)^{++k}) b)$ and $\iota(k, b)$ by something more general:

$$\bullet y_1, y_2 : C_{\infty}(n+1, a(n, a)) \text{ and } \beta : y_1 = y_2$$

We replace $\mu(n, a, k+1, b((n, a)^{++k}) b)$ and $\mu(n, a, k, b)$ by arbitrary paths

$$\begin{aligned} \bullet u_1 : g(n, a, c(n, a) y_1) = \\ \Delta(\kappa(n, a), c_{\star}(n, a), y_1) \#_E g(n+1, a(n, a), y_1) \\ \bullet u_2 : g(n, a, c(n, a) y_2) = \\ \Delta(\kappa(n, a), c_{\star}(n, a), y_2) \#_E g(n+1, a(n, a), y_2) \end{aligned}$$

But the path $\kappa(k, b)$ was not arbitrary at all: we had the further piece of information that

$$\text{ap}_{c(n, a)}(\kappa(k, b)) = \text{ap}_{\iota(k+2)}(\delta(n, a, k, b)) \cdot \kappa(k+1, Q(n, a, k) b)$$

This is just the second computation rule of $c(n, a)$, the witness for which we call $\beta_c(n, a, k, b)$. We capture this abstractly as having:

$$\begin{aligned} \bullet z : C_{\infty}(n, a) \\ \bullet \gamma_1 : c(n, a) y_1 = z \\ \bullet \gamma_2 : z = c(n, a) y_2 \\ \bullet \theta : \text{ap}_{c(n, a)}(\beta) = \gamma_1 \cdot \gamma_2 \end{aligned}$$

In this abstract scenario, we want to relate u_1 and u_2 over the path β . Since u_1 is a path in $E(\iota(n, a), c(n, a) y_1)$ and u_2

$$\begin{array}{ccc}
\gamma_2 \#_{E(\iota(n,a),-)} \gamma_1 \#_{E(\iota(n,a),-)} \Delta(\kappa(n,a), c_\star(n,a), y_1) \#_E g(n+1, a(n,a), y_1) & \xrightarrow{??} & \Delta(\kappa(n,a), c_\star(n,a), y_2) \#_E g(n+1, a(n,a), y_2) \\
\uparrow \text{ap}_{\gamma_2 \#_{E(\iota(n,a),-)}} (\text{ap}_{\gamma_1 \#_{E(\iota(n,a),-)}} (u_1)) & & \uparrow u_2 \\
\gamma_2 \#_{E(\iota(n,a),-)} \gamma_1 \#_{E(\iota(n,a),-)} g(n, a, c_\star(n,a) y_1) & \xrightarrow{??} & g(n, a, c_\star(n,a) y_2)
\end{array}$$

Figure 1. Filling the diagram

is a path in $E(\iota(n,a), c(n,a) y_2)$, to relate them we first need to have them in the same fiber of E . The two fibers of E can be related by the transport of the type family $E(\iota(n,a), -)$ over $\text{ap}_{c(n,a)}(\beta)$, which we could use to carry u_1 over. But since we know $\text{ap}_{c(n,a)}(\beta) = \gamma_1 \cdot \gamma_2$, it is better to use the transports over γ_1 and then γ_2 in succession. So we want to fill the diagram in figure 1. Using dependent application of g , we can fill edges as shown in figure 2. The remaining edge, which we call ε , can be seen as witnessing the naturality of $\Delta(\kappa(n,a), c_\star(n,a))$ and is defined simultaneously with proving that the commutativity of the diagram in figure 2 implies that u_1 and u_2 are suitably related over β .

Putting all this together, we have the following generalization: given:

- $y_1, y_2 : C_\infty(n+1, a(n,a))$ and $\beta : y_1 = y_2$
- $z : C_\infty(n,a)$
- $\gamma_1 : c(n,a) y_1 = z$
- $\gamma_2 : z = c(n,a) y_2$
- $\theta : \text{ap}_{c(n,a)}(\beta) = \gamma_1 \cdot \gamma_2$

there is a path ε as indicated in figure 2 such that for any

- $u_1 : g(n, a, c(n,a) y_1) = \Delta(\kappa(n,a), c_\star(n,a), y_1) \#_E g(n+1, a(n,a), y_1)$
- $u_2 : g(n, a, c(n,a) y_2) = \Delta(\kappa(n,a), c_\star(n,a), y_2) \#_E g(n+1, a(n,a), y_2)$

we have the equivalence in figure 3. To show this, we proceed as follows:

- First we do path induction on β , to collapse y_1 and y_2 .
- We then perform path induction on γ_1 , which is possible since γ_1 has one free endpoint z . This replaces c with $c(n,a) y_1$. We cannot perform induction on γ_2 right away since both of its endpoints are now fixed to $c(n,a) y_1$.
- However, we still have the coherence θ , which now has type $\text{refl} = \text{refl} \cdot \gamma_2$. Since θ itself does not appear in what we are trying to prove, we can simply replace its type with something equivalent, namely $\text{refl} = \gamma_2$.
- The new θ now has one free endpoint, which means we can perform induction on it, reducing γ_2 to reflexivity.

After carrying out this procedure, the two vertices of ε both reduce to

$$\Delta(\kappa(n,a), c_\star(n,a), y_1) \#_E g(n+1, a(n,a), y_1)$$

so this edge can just be reflexivity. In the remaining part, the goal has reduced to showing that $u_1 = u_2$ is equivalent to $\text{refl} = \text{refl} \cdot \text{ap}_{\text{id}}(\text{ap}_{\text{id}}(u_1)) \cdot \text{refl} \cdot \text{refl} \cdot u_2^{-1}$ and this is clearly the case. We will refer to the edge we just constructed as $\varepsilon(\beta, \gamma_1, \gamma_2, \theta)$. Our intended instantiation is thus:

- $y_1 := \iota(k+1, b((n,a)^{++k}) b)$
- $y_2 := \iota(k, b)$
- $\beta := \kappa(k, b)$
- $z := \iota(k+2, b((n,a)^{k+1}) Q(n, a, k, b))$
- $\gamma_1 := \text{ap}_{\iota(k+2)}(\delta(n, a, k, b))$
- $\gamma_2 := \kappa(k+1, Q(n, a, k, b))$
- $\theta := \beta_c(n, a, k, b)$
- $u_1 := \mu(n, a, k+1, b((n,a)^{++k}) b)$
- $u_2 := \mu(n, a, k, b)$

So to prove 3 it suffices to establish the equality in figure 4. On the left hand side, the path $\text{apd}_{g(n,a)}(\kappa(k+1, Q(n, a, k, b)))$ is equal to $H(n, a, k+1, Q(n, a, k, b))$; on the right hand side, the path $\text{apd}_{g(n+1, a(n,a))}(\kappa(k, b))$ is equal to $H(n+1, a(n,a), k, b)$. Thus, we have essentially expressed $H(\cdot, \cdot, k+1, \cdot)$ in terms of $H(\cdot, \cdot, k, \cdot)$, which gives us the successor step of the definition of H by induction (the most difficult part of the proof). For the zero case, and the verification that these definitions indeed satisfy the intended computation rules, we refer the reader to the technical report accompanying the submission.

7 Applications of the Main Theorem

We will need a couple more properties of colimits before we can look at applications of the main theorem. We recall that given a sequence (A, a) we have a function $(-)^{+k} : A(n) \rightarrow A(n+k)$. Sometimes the codomain we need is not exactly of the form $A(n+k)$. For example, if $a : A(0)$, then $(0, a)^{+k}$ has type $A(0+k)$ which is not definitionally equal to $A(k)$. So, generally, given a proof that $n \leq m$ we define $a^{n \leq m} : A(n) \rightarrow A(m)$ inductively by taking $a^{n \leq n} := \text{id}_{A_n}$ and $a^{n \leq m+1} := a_m \circ a^{n \leq m}$. We will omit the proof that $n \leq m$ as an argument, which is fine to do since an inequality on \mathbb{N} is a mere proposition. It is easy to show that $a^{n \leq n+k}(a) = (n, a)^{+k}$.

The following lemma states that $\iota(0)$ is an equivalence if all maps in the sequence are an equivalence. We will have a more general result in Corollary 7.6.5, but in that proof we will use a special case of this lemma. We will omit the proof of this lemma here.

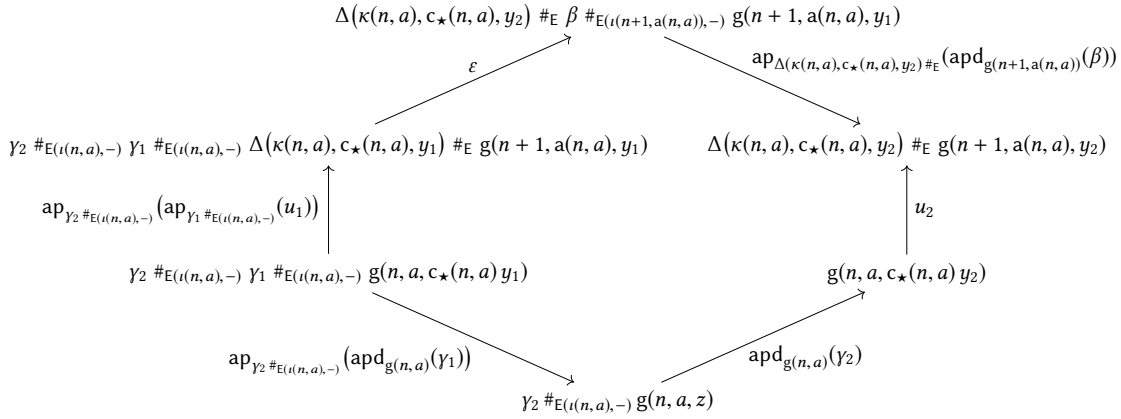


Figure 2. Filling the diagram, continued

$$\begin{aligned}
& \beta \#_{y \rightarrow g(n, a, c(n, a), y) = \Delta(\kappa(n, a), c_*(n, a), y) \#_E g(n+1, a(n, a), y)} u_1 = u_2 \\
& \simeq \\
& \text{apd}_{g(n, a)}(Y_2) = \text{ap}_{Y_2 \#_{E(i(n, a), -)}} \left(\text{apd}_{g(n, a)}(Y_1) \right)^{-1} \cdot \text{ap}_{Y_2 \#_{E(i(n, a), -)}} \left(\text{ap}_{Y_1 \#_{E(i(n, a), -)}}(u_1) \right) \cdot \varepsilon \cdot \\
& \quad \text{ap}_{\Delta(\kappa(n, a), c_*(n, a), y_2) \#_E} \left(\text{apd}_{g(n+1, a(n, a))}(\beta) \right) \cdot u_2
\end{aligned}$$

Figure 3. Equality of u_1 and u_2 over β is the same as the commutativity of the diagram in figure 2

$$\begin{aligned}
& \text{apd}_{g(n, a)} \left(\kappa(k+1, Q(n, a, k, b)) \right) = \\
& \quad \text{ap}_{\kappa(k+1, Q(n, a, k, b)) \#_{E(i(n, a), -)}} \left(\text{apd}_{g(n, a)} \left(\text{ap}_{i(k+2)}(\delta(n, a, k, b)) \right) \right)^{-1} \cdot \\
& \quad \text{ap}_{\kappa(k+1, Q(n, a, k, b)) \#_{E(i(n, a), -)}} \left(\text{ap}_{\text{ap}_{i(k+2)}(\delta(n, a, k, b)) \#_{E(i(n, a), -)}} \left(\mu(n, a, k+1, b((n, a)^{++k}, b)) \right) \right) \cdot \\
& \quad \varepsilon \left(\kappa(k, b), \text{ap}_{i(k+2)}(\delta(n, a, k, b)), \kappa(k+1, Q(n, a, k, b)), \beta_c(n, a, k, b) \right) \cdot \\
& \quad \text{ap}_{\Delta(\kappa(n, a), c_*(n, a), i(k, b)) \#_E} \left(\text{apd}_{g(n+1, a(n, a))}(\kappa(k, b)) \right) \cdot \mu(n, a, k, b)^{-1}
\end{aligned}$$

Figure 4. Our new goal

Lemma 7.1. Given a sequence (A, a) where $a(n)$ is an equivalence for all n , then the map $i(0) : A(0) \rightarrow A_\infty$ is an equivalence.

We also recall the following lemma:

Lemma 7.2. Let $a : A$, and let $B : A \rightarrow \mathcal{U}$ be a type family with $b : B(a)$. Then the following are equivalent:

1. The canonical family of maps $\Pi_{x:A}(a = x) \rightarrow B(x)$ is a family of equivalences.
2. The total space of B is contractible.

In our first application we show that sequential colimits commute with the identity type.

Theorem 7.3. Consider a sequence (A, a) . Then for any $a, a' : A(n)$ there is an equivalence

$$(i(n, a) =_{A_\infty} i(n, a')) \simeq \text{colim}_k ((n, a)^{+k} = (n, a')^{+k}).$$

Proof. We first prove this for $n \equiv 0$. Note that for any $a_0 : A(0)$, we have a fibered type sequence over (A, a) given by $B(n, a) := a^{0 \leq n}(a_0) =_{A(n)} a$ and $b(n, a, p) = \text{ap}_a(p)$. Now we use Theorem 5.1 to see that the total space of B_∞ is contractible.

$$\begin{aligned}
\Sigma_{a:A_\infty} B_\infty(a) & \simeq \text{colim}_n (\Sigma_{a:A(n)} a^{0 \leq n}(a_0) = a) \\
& \simeq \text{colim}_n (1) \\
& \simeq 1.
\end{aligned}$$

Since $\iota(0, \text{refl}_{a_0}) : B_\infty(\iota(0, a_0))$ we can now conclude by Lemma 7.2 that

$$\begin{aligned} & (\iota(0, a_0) =_{A_\infty} \iota(0, a)) \\ & \simeq B_\infty(\iota(0, a)) \\ & \equiv \text{colim}((0, a_0)^{0 \leq 0+k} =_{A(0+k)} (0, a)^{+k}) \\ & \simeq \text{colim}((0, a_0)^{+k} =_{A(0+k)} (0, a)^{+k}). \end{aligned}$$

The last step uses that $a^{0 \leq 0+k}(a_0) = (0, a_0)^{+k}$.

For general n , we denote the sequence $(k \mapsto A(n+k), k \mapsto a(n+k))$ as $S_n(A, a)$ or $(S_n(A), S_n(a))$. By iterating Lemma 3.5 we get an equivalence $e : A_\infty \simeq (S_n(A, a))_\infty$. For $a, a' : A(n)$ we can now compute:

$$\begin{aligned} & (\iota(n, a) =_{A_\infty} \iota(n, a')) \\ & \simeq (e_n(\iota(n, a)) =_{\text{colim}(S_n(A, a))} e_n(\iota(n, a'))) \\ & \simeq (\iota(0, a) =_{\text{colim}(S_n(A, a))} \iota(0, a')) \\ & \simeq \text{colim}((0, a)^{+k} =_{S_n(A)_{0+k}} (0, a')^{+k}) \\ & \simeq \text{colim}((n, a)^{+k} =_{A(n+k)} (n, a')^{+k}). \end{aligned}$$

The last equivalence comes from a natural isomorphism of the sequences, because there is a dependent path between $S_n(f)^k(a)$ and $(n, a)^{+k}$ over the canonical path that $n + (0 + k) = n + k$. \square

A *pointed type* A is a pair consisting of a type, usually also denoted A and a basepoint $a_0 : A$. A *pointed map* $f : A \rightarrow^* B$ between pointed types is a pair consisting of a map $f : A \rightarrow B$ and a path $f_0 : f(a_0) = b_0$. The *loop space* ΩA of a pointed type A is a pointed type with underlying type $a_0 = a_0$ and basepoint refl_{a_0} . A pointed map $f : A \rightarrow^* B$ induces a pointed map $\Omega f : \Omega A \rightarrow^* \Omega B$ which has as underlying map $\Omega f(p) := f_0^{-1} \cdot \text{ap}_f(p) \cdot f_0$.

A *pointed type sequence* is a type sequence where all types and maps are pointed. We will denote the basepoint of A_n by a_n , and now the sequential colimit is pointed with basepoint $\iota(0, a_0)$. The next corollary shows that colimits commute with taking the loop space. This proves Conjecture 3.8 of [Cavallo 2015] and answers Question 4.2 of [Graham 2017].

Corollary 7.4. *For a pointed type sequence (A, a) we have the following pointed equivalence*

$$\Omega A_\infty \simeq \text{colim}(n \mapsto \Omega A(n), n \mapsto \Omega a(n)).$$

Proof. We compute

$$\begin{aligned} \Omega A_\infty & \simeq \text{colim}_n((0, a_0)^{+n} = (0, a_0)^{+n}) \\ & \simeq \text{colim}_n(a^{0 \leq n}(0, a_0) = a^{0 \leq n}(0, a_0)) \\ & \simeq \text{colim}_n(n \mapsto \Omega A(n), n \mapsto \Omega a(n)). \end{aligned}$$

The first equivalence is Theorem 7.3, and the last two are by functoriality of the colimit. The second equivalence uses that $(0 + n, (0, a_0)^{+n}) = (n, a^{0 \leq n}(a_0))$ and the third uses that $a^{0 \leq n}(0, a_0) = a_n$, both of which are easily proven by induction on n . The naturality of these equalities is also easily

proven by induction on n . The fact that this equivalence is pointed is by reflexivity. \square

Given a natural transformation $\tau : (A', a') \rightarrow (A, a)$ we can define the homotopy fiber of τ as a type sequence fibered over (A, a) , by $\text{fib}_\tau(n, a) := \text{fib}_{\tau(n)}(a)$. Then the following corollary of the main theorem states that colimits commute with homotopy fibers.

Corollary 7.5. *Given a natural transformation $\tau : (A', a') \rightarrow (A, a)$ and a point $x : A_\infty$. Then*

$$\text{fib}_{\tau_\infty}(x) \simeq (\text{fib}_\tau)_\infty(x).$$

Proof. Consider the following diagram, where the equivalences on the top are given by Theorem 5.1 and the fact that the total space of the fiber of a function is the domain of that function.

$$\begin{array}{ccc} \Sigma_{x:A_\infty}(\text{fib}_\tau)_\infty(x) & \xrightarrow{\sim} & \text{colim}_k(\Sigma_{x:A(n)}\text{fib}_{\tau(n)}(x)) & \xrightarrow{\sim} & A'_\infty \\ & \searrow \pi_1 & \downarrow p & \swarrow \tau_\infty & \\ & & A_\infty & & \end{array}$$

In this diagram p is induced by taking the first projection. This diagram commutes: the left triangle commutes by Theorem 5.1 and the right triangle commutes by the 1-functoriality of the colimit, Lemma 3.7. Therefore,

$$\text{fib}_{\tau_\infty}(x) \simeq \text{fib}_{\pi_1}(x) \simeq (\text{fib}_\tau)_\infty(x). \quad \square$$

The following corollary specifies the interaction of colimits and truncations, truncatedness and connectedness. In particular, we show that colimits of n -truncated types is again n -truncated. In Theorem 3.3 of [Graham 2017] the definition of a homology theory is given as the set-truncation of a colimit of stable homotopy groups. Our result shows that this set-truncation is superfluous: the colimit is already a set.

Corollary 7.6. *Consider a sequence (A, a) and some $k \geq -2$.*

1. *If $A(n)$ is k -truncated for all $n : \mathbb{N}$, then A_∞ is k -truncated.*
2. *We have an equivalence*

$$\|A_\infty\|_k \simeq \text{colim}_n(\|A(n)\|_k, \|a(n)\|_k).$$
3. *If $A(n)$ is k -connected for all $n : \mathbb{N}$, then A_∞ is k -connected.*
4. *Given a natural transformation $(\tau, H) : (A, a) \rightarrow (A', a')$ such that $\tau(n)$ is k -truncated (k -connected) for all n , then τ_∞ is k -truncated (k -connected).*
5. *If $a(n)$ is k -truncated (k -connected) for all n , then $\iota(0)$ is k -truncated (k -connected).*

B y Lemma 3.5 we can generalize the quantification “for all $n : \mathbb{N}$ ” in this Corollary to the weaker “there is an $m : \mathbb{N}$ such that for all $n \geq m$ ”. In part 5 the conclusion then becomes that $\iota(m)$ is k -truncated (k -connected).

Proof (of Corollary 7.6).

1. We prove this by induction on k . Suppose $k = -2$, then $A(n)$ is an equivalence for all n . Therefore $A_\infty \simeq A(0)$ by Lemma 7.1, hence A_∞ is contractible.

Now suppose $k \equiv k' + 1$. Take $x, x' : A_\infty$, we need to show that $x = x'$ is k' -truncated. Since being truncated is a mere proposition, by induction on x and x' we may assume that $x \equiv \iota(n, a)$ and $x' \equiv \iota(m, a')$. Now $\iota(n, a) = \iota(n+m, a^{n \leq n+m}(a))$ and $\iota(m, a') = \iota(n+m, a^{m \leq n+m}(a'))$, therefore the type $\iota(n, a) = \iota(m, a')$ is equivalent to

$$\iota(n+m, a^{n \leq n+m}(a)) = \iota(n+m, a^{m \leq n+m}(a')).$$

Therefore it suffices to show that the latter equality type is k' -truncated. By Theorem 7.3 we need to show that the type

$$\text{colim}_\ell((n+m, a^{n \leq n+m}(a))^{+\ell} = (n+m, a^{m \leq n+m}(a'))^{+\ell})$$

is k' -truncated, which follows by the induction principle and the fact that $A(n+m+\ell)$ is $(k'+1)$ -truncated.

2. By the functoriality of the sequential colimit, we have a map $A_\infty \rightarrow \text{colim}_n(\|A(n)\|_k, \|a(n)\|_k)$. Because the right hand side is k -truncated, this induces a map

$$g : \|A_\infty\|_k \rightarrow \text{colim}_n(\|A(n)\|_k, \|a(n)\|_k).$$

For the other direction, we define a map

$$h : \text{colim}_n(\|A(n)\|_k, \|a(n)\|_k) \rightarrow \|A_\infty\|_k$$

by

$$\begin{aligned} \iota(n, |a|_k) &\mapsto |\iota(n, a)|_k \\ \kappa(n, |a|_k) &\mapsto \text{ap}_{|-|_k}(\kappa(n, a)) \end{aligned}$$

It is straightforward to show that both $h \circ g$ and $g \circ h$ are homotopic to the identity.

3. Since $A(n)$ is k -connected, $\|A(n)\|_k$ is contractible, and therefore $\|A_\infty\|_k \simeq \text{colim}_n(\|A(n)\|_k)$ is contractible.
4. Let $x : A'_\infty$. We need to show that $\text{fib}_{\tau_\infty}(x)$ is k -truncated (k -connected). This is a mere proposition, so we may assume that $x \equiv \iota(n, a)$ for some $a : A(n)$. Now

$$\text{fib}_{\tau_\infty}(\iota(n, a)) \simeq (\text{fib}_\tau)_\infty(\iota(n, a))$$

by Corollary 7.5. Since $\text{fib}_{\tau(n)}(x)$ is k -truncated (k -connected) for all n and x , we know that its colimit $(\text{fib}_\tau)_\infty(\iota(n, a))$ is k -truncated (k -connected), by part 1 or 3. This finished the proof.

5. Consider the natural transformation

$$\begin{array}{ccccccc} A(0) & \xlongequal{\quad} & A(0) & \xlongequal{\quad} & \cdots & \xlongequal{\quad} & \text{colim}_n(A(0)) \\ \parallel_{a^{0 \leq 0}} & & \parallel_{a^{0 \leq 1}} & & & & \downarrow \\ A(0) & \longrightarrow & A(1) & \longrightarrow & \cdots & \longrightarrow & A_\infty \end{array}$$

The maps $a^{0 \leq n} : A(0) \rightarrow A(n)$ are k -truncated (k -connected) and form a natural transformation. Therefore, by part 4 the map $a^{0 \leq \infty} : \text{colim}_n(A(0)) \rightarrow A_\infty$ is k -truncated (k -connected). The fiber of $\iota(0)$ over $x : A_\infty$

is the same as the fiber of $a^{0 \leq \infty}$ over x , and therefore $\iota(0)$ is k -truncated (k -connected). \square

8 Conclusion

We presented here the original proof of the conjecture that in homotopy type theory sequential colimits commute with Σ -types. In our proof, sequential colimits are formulated as a higher inductive type and we reason directly by the associated induction principle (together with path induction). By contrast, Bocquet's independent proof [Bocquet 2017] of the analogous result for cubical type theory uses an alternative formulation of sequential colimits as an *indexed* higher inductive type. This indexed HIT has constructors $\iota(x)$ and $\kappa(x)$ for $x : A(0)$ (i.e., not for a general $n : \mathbb{N}$ but only for the base case $n := 0$), plus an additional constructor lift that takes an element $x : \text{colim}(n \mapsto A(n+1), n \mapsto a(n+1))$ of the colimit of the *lifted* type sequence that drops the first index, and produces a canonical element of the colimit of the original type sequence. The type family B_∞ as well as the two inverse maps witnessing the desired equivalence between the colimit of the sums and the sum of colimits are then constructed using the alternative induction principle associated with the indexed HIT. In this approach, one only distinguishes the base case $\iota(x)$ and the inductive case $\text{lf}(x)$, which eliminates the need for reasoning about k -fold liftings or natural transformations (section 3) in the proof of the main theorem – however, we still need to develop these for our corollaries. The second proof by Bocquet follows essentially the same line of reasoning, using Agda coinductive types to mimic the behavior of the indexed HIT.

We opted to carry out our proofs in homotopy type theory as opposed to a version of cubical type theory for several reasons: the conjecture as originally formulated by Rijke was for homotopy type theory; we prefer to keep the assumptions on the underlying type theory minimal; since cubical type theories interpret homotopy type theory, our proof can readily be adapted to any of these alternative systems; and, finally, the extra judgmental equalities for higher constructors provided by cubical type theory do not result in any significant simplification of the proof (each of the typical computation rules corresponds to a simple rewrite).

We have used the proof assistant Lean 2¹ to formalize our results. Lean 2 is an older version of Lean² that offers significantly more support for HoTT [de Moura et al. 2015]. We include the formalization as part of the submission. The formalization closely follows the proofs of this paper, with some differences in notation and setup due to the use of cubical techniques like pathovers and squareovers [Licata and Brunerie 2015] in the implementation.

¹<https://github.com/leanprover/lean2>

²<https://leanprover.github.io/>

References

- 1321 Agda Development Team. 2019. Coinduction. (2019). Agda user manual.
1322 Available at <https://agda.readthedocs.io/en/v2.5.2/language/coinduction.html>.
1323
1324
- 1325 B. Ahrens, P. Capriotti, and R. Spadotti. 2015. Non-Wellfounded Trees in
1326 Homotopy Type Theory. In *Typed Lambda Calculi and Applications (TLCA 2015)*. Vol. 38. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern,
1327 17–30.
- 1328 S. Awodey, R. Garner, P. Martin-Löf, and V. Voevodsky (organizers). 2011.
1329 Mini-Workshop (Oberwolfach): The Homotopy Interpretation of Con-
1330 structive Type Theory. (2011). Available at https://hottheory.files.wordpress.com/2011/06/report-11_2011.pdf.
1331
- 1332 S. Awodey and M. Warren. 2009. Homotopy-Theoretic Models of Identity
1333 Types. *Mathematical Proceedings of the Cambridge Philosophical Society*
1334 146, 1 (2009), 45–55.
- 1335 R. Bocquet. 2017. Proof of the Commutativity of Sums and Sequential
1336 Colimits in Cubical Type Theory. (2017). Available at <https://github.com/mortberg/cubicaltt/blob/seqcolim/examples/seqcolim.ctt>.
1337
- 1338 R. Bocquet. 2018. Coinductive Sequential Colimits. (2018). Available at
1339 <https://github.com/RafaelBocquet/CoinductiveSequentialColimits>.
1340
- 1341 G. Brunerie. 2016. On the Homotopy Groups of Spheres in Homotopy Type
1342 Theory. (2016). Available at [arxiv.org as arXiv:1606.05916](https://arxiv.org/abs/1606.05916).
1343
- 1344 E. Cavallo. 2015. *Synthetic Cohomology in Homotopy Type Theory*. Master's
1345 thesis. Carnegie Mellon University.
- 1346 J. Christensen, M. Opie, E. Rijke, and L. Scoccola. 2018. Localization in
1347 Homotopy Type Theory. (2018).
1348
- 1349 C. Cohen, T. Coquand, S. Huber, and A. Mörtberg. 2016. Cubical Type
1350 Theory. (2016). Available at <https://github.com/mortberg/cubicaltt>.
1351
- 1352 L. de Moura, S. Kong, J. Avigad, F. van Doorn, and J. von Raumer. 2015. The
1353 Lean Theorem Prover (System Description). *Conference on Automated
1354 Deduction (CADE 2015)* (2015), 378–388.
1355
- 1356 R. Graham. 2017. Synthetic Homology in Homotopy Type Theory. (2017).
1357 Available at [arxiv.org as arXiv:1706.01540](https://arxiv.org/abs/1706.01540).
1358
- 1359 P. Hirschhorn. 2019. Notes on Homotopy Colimits and Homotopy Limits.
1360 (2019). Available at <http://www-math.mit.edu/~psh/notes/hocolim.pdf>.
1361
- 1362 C. Kapulkin and P. LeFanu Lumsdaine. 2012. The Simplicial Model of Univa-
1363 lent Foundations (after Voevodsky). (2012). To appear in the *Journal of Eu-
1364 ropean Mathematical Society*. Available at [arxiv.org as arXiv:1203.3253v5](https://arxiv.org/abs/1203.3253v5).
1365
- 1366 N. Kraus and J. von Raumer. 2019. Path Spaces of Higher Inductive Types
1367 in Homotopy Type Theory. In *To appear in Logic in Computer Science
1368 (LICS 2019)*.
1369
- 1370 D. Licata and G. Brunerie. 2015. A Cubical Approach to Synthetic Homotopy
1371 Theory. In *Logic in Computer Science (LICS 2015)*. IEEE, 92–103.
1372
- 1373 P. LeFanu Lumsdaine. 2011. Higher Inductive Types: a Tour of the Menagerie.
1374 (2011). Available at <https://homotopytypetheory.org/2011/04/24/higher-inductive-types-a-tour-of-the-menagerie/>.
1375
- 1376 P. LeFanu Lumsdaine and M. Shulman. 2017. Semantics of Higher Inductive
1377 Types. (2017). Available at [arxiv.org as arXiv:1705.07088v2](https://arxiv.org/abs/1705.07088v2).
1378
- 1379 P. Martin-Löf. 1975. An Intuitionistic Theory of Types: Predicative Part. In
1380 *Logic Colloquium 1973*. North-Holland.
1381
- 1382 E. Rijke. 2017. The Join Construction. (2017). Available at [arxiv.org as arXiv:1701.07538](https://arxiv.org/abs/1701.07538).
1383
- 1384 M. Shulman. 2011a. Homotopy Type Theory, VI. Available at https://golem.ph.utexas.edu/category/2011/04/homotopy_type_theory_vi.html.
1385 (2011).
1386
- 1387 M. Shulman. 2011b. An interval type implies function extensionality. (2011).
1388 Post on the Homotopy Type Theory blog.
1389
- 1390 M. Shulman. 2013. Spectral Sequences. Available at <https://homotopytypetheory.org/2013/08/08/spectral-sequences/>. (2013).
1391
- 1392 M. Shulman. 2019. All $(\infty, 1)$ -Toposes Have Strict Univalent Universes.
1393 (2019). Available at [arxiv.org as arXiv:1904.07004](https://arxiv.org/abs/1904.07004).
1394
- 1395 K. Sojakova. 2016. *Higher Inductive Types as Homotopy-Initial Algebras*.
1396 Ph.D. Dissertation. Carnegie Mellon University.
1397
- 1398 The Univalent Foundations Program, Institute for Advanced Study. 2013. *1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430*
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430