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 (by Hirschorn) from topological spaces to arbitrary sequential colimits. Moreover, using Shulman’s recent interpretation of homotopy type theory into (∞, 1)-toposes, we are able to extend the classical analogue of our main result (by Hirschorn) from topological spaces to arbitrary (∞, 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 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

\[ A(0) \xrightarrow{a(0)} A(1) \xrightarrow{a(1)} A(2) \xrightarrow{a(2)} \cdots \]  

(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 \( \Lambda_\infty \). The point constructor of \( \Lambda_\infty \) is

\[ i : \Pi_{n : \mathbb{N}} A(n) \rightarrow \Lambda_\infty, \]

and the path constructor of \( \Lambda_\infty \) is

\[ \kappa : \Pi_{n : \mathbb{N}} \Pi_{x, y : A(n)} \tilde{\iota}(n + 1, a(n, x)) = i(n, x). \]

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 \( \Lambda_\infty \) indeed behaves as the homotopy colimit of the type sequence \( A(0) \rightarrow A(1) \rightarrow \cdots \). 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 \( B(n) : \Pi_{a : \mathbb{N}} \rightarrow U \) 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:

\[ \begin{array}{c}
B(0) \\
\downarrow b(0)
\end{array} \quad \begin{array}{c}
B(1) \\
\downarrow b(1)
\end{array} \quad \begin{array}{c}
B(2) \\
\downarrow b(2)
\end{array} \cdots \\
\begin{array}{c}
A(0) \\
\downarrow a(0)
\end{array} \quad \begin{array}{c}
A(1) \\
\downarrow a(1)
\end{array} \quad \begin{array}{c}
A(2) \\
\downarrow a(2)
\end{array} \cdots \]
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. $\Sigma$-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 $B_\infty : A_\infty \to U$ and compare its total space with $\text{colim}(\Sigma(A, B))$. To construct $B_\infty$, 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 $B_\infty : A_\infty \to U$ as a result. Our main technical theorem, Theorem 5.1, states that we have a commuting triangle

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

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 (∞, 1)-
toposes. Our result in homotopy type theory thus generalizes
Hirschhorn’s theorem to arbitrary (∞, 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 concep-
tual 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 cer-
tain typal 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 ho-
motopy 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 fam-
ily Bο; in section 5 we state our main result and show that
it follows from a more convenient induction principle on
ΣτθBο(τ); in section 6 we sketch the proof of the afore-
mentioned 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 ho-
motopy 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 homot-
opy type theory: judgmental equality and typal equality.
Judgmental equality is used for computation. When two ex-
pressions t₁ and t₂ for terms of type A describe the exact
same term, we write t₁ ≡ t₂. For example, 1 + 1 ≡ 2 of type
N. The second form of equality is typal 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
reflₓ. For any two paths p : x = y and q : y = z, we write p ⋆ q
for their concatenation. Concatenation is defined in such a
way that reflₓ ⋆ reflₓ ≡ reflₓ, and there is an identification
(p ⋆ q) ⋆ r ≡ p ⋆ (q ⋆ r) for any three concatenable paths.
Moreover, any function f : A → B respects paths: for any
path p : x = y in A there is a path apₓ(f)(p) : f(x) = f(y).

We recall that a homotopy between two (dependent) maps
f, g : ΠₓA(B(x)) is a term of type f ≅ g : ΠₓA(f(x)) = g(x).
An equivalence is then defined as a map f : A → B equipped
with a right and a left inverse, i.e. maps g : B → A and
h : B → A with homotopies f ⋆ g ∼ idₓB and h ⋆ f ∼ idₓA. We
write A ≃ B for the type of equivalences from A to B. The
univalence axiom asserts that the canonical map (A ≃ B) →
(A ≃ B) is an equivalence. We will write p : A ≃ B for the
function associated to a path p : A = B in U. The function
extensionality principle (which follows from univalence)
asserts that for any f, g : ΠₓA B(x), the map (f = g) → (f ≃
g) is an equivalence. For f : ΠₓA B and y : B, the homotopy
fiber of f at y is defined as fibₓy(y) := ΠₓA f(x) = y.

An important concept in homotopy type theory is that of
transport. Given a type family B : A → U and a path
p : x = y in A, we have a function p#B : B(x) → B(y), with
reflₓ #B ≡ id. Using transport, we can also show that depend-
ent functions respect paths: for any dependent function
f : ΠₓA 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 apₓ(p) : p#B(f(x)) = f(y) in B(y).

The notion that a type A is n-truncated is defined by re-
cursion on n : ℕ, n ≥ −2. We say that A is (−2)-truncated
or contractible if we can find a inhabitant of Στθ AΠτy A = 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∥−2 is an n-truncated type equipped with
a map ∥−∥n : A → ∥A∥n such that for any n-truncated type
B the precomposition map (∥A∥n) → B → (A → B) is an
equivalence. A type A is n-connected if ∥A∥n is contractible.
A function f : ΠₓA B is called n-truncated (n-connected)
if for all y : B the type fibₓ(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) : Seq with

• A : ℕ → U

• a : ΠnA(n) → 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) → (A′, a′) is a pair (τ, H)
consisting of a family of maps

τ : ΠnA(n) → A′(n)

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

\[
\begin{array}{cccccc}
A(0) & \to & A(1) & \to & A(2) & \to & \cdots \\
\downarrow \tau(0) & & \downarrow \tau(1) & & \downarrow \tau(2) \\
A′(0) & \to & A′(1) & \to & A′(2) & \to & \cdots
\end{array}
\]

commutes, i.e. Hₙ : τ(n + 1) ◦ a(n) ~ a′(n) ◦ τ(n).
A natural equivalence is a natural transformation \((\tau, H)\) 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:

\[
\begin{align*}
\iota & : \Pi_{n:N}(A(n) \to \text{colim}(A, a)) \\
\kappa & : \Pi_{n:N}(A_{n+1}(n+1, a(n, a)) = \iota(n, a)
\end{align*}
\]

The constructor \(\iota\) gives the canonical injection of \(a : A(n)\) into the colimit, whereas the constructor \(\kappa\) provides the glueing 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 typall equal. The higher dimensionality of this inductively-defined type comes from the fact that \(\kappa\) 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 \(A_\infty\) if the maps can be inferred from the context.

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

\[
\begin{align*}
E & : \text{colim}(A, a) \to U \\
e & : \Pi_{n:N} E(A(n)) \\
p & : \Pi_{n:N} \Pi_{a:A(n)} E(e(n, a))
\end{align*}
\]

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

\[
\begin{align*}
\text{ind}(E, e, p) \iota(n, a) & \equiv e(n, a) \\
\text{ap}_{\text{ind}(E, e, p)}(\kappa(n, a)) & \equiv p(n, a)
\end{align*}
\]

The induction principle implies the following recursion principle: given

\[
\begin{align*}
E & : U \\
e & : \Pi_{n:N} E(A(n)) \\
p & : \Pi_{n:N} \Pi_{a:A(n)} e(n + 1, a(n, a)) = e(n, a)
\end{align*}
\]

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

\[
\begin{align*}
\text{rec}(E, e, p) \iota(n, a) & \equiv e(n, a) \\
\text{ap}_{\text{rec}(E, e, p)}(\kappa(n, a)) & \equiv p(n, a)
\end{align*}
\]

One important consequence of the induction principle is:

**Lemma 3.4 (Uniqueness property of the sequential colimit).**

Two functions \(F_1, F_2 : \text{colim}(A, a) \to E\) out of the sequential colimit are equal if there is a homotopy

\[
\alpha : \Pi_{n:N} \Pi_{a:A(n)} F_1(i(n, a)) = F_2(i(n, a))
\]

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

\[
\begin{align*}
F_1(i(n + 1, a(n, a))) & \quad \text{ap}_{F_1}(\kappa(n, a)) & \quad F_1(i(n, a)) \\
\alpha(n + 1, a(n, a)) & \quad F_1(i(n, a)) & \quad \alpha(n, a)
\end{align*}
\]

\[
\begin{align*}
F_2(i(n + 1, a(n, a))) & \quad \text{ap}_{F_2}(\kappa(n, a)) & \quad F_2(i(n, a)) \\
\alpha(n + 1, a(n, a)) & \quad F_2(i(n, a)) & \quad \alpha(n, a)
\end{align*}
\]

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

\[
A(1) \to A(2) \to A(3) \to \cdots
\]

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

\[
i(n, a) \mapsto i(n + 1, a) \\
\kappa(n, a) \mapsto \kappa(n + 1, a)
\]

**Proof.** We define the map in the opposite direction by

\[
i(n, a) \mapsto i(n, a(n, a)) \\
\kappa(n, a) \mapsto \kappa(n, a(n, a))
\]

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

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 \(\Sigma A := \sum_{n:N} A(n)\) and define the lifting \((\cdot)^* : \Sigma A \to \Sigma A\) by \((n, a)^* := (n + 1, a(n, a))\). We define the \(k\)-fold lifting \((\cdot)^{k+1} : \Sigma A \to \Sigma A\) inductively by \(x^{0} := x\) and \(x^{k+1} := (x^{k})^{*}\). Lifting is associative up to typal equality: we define \(q(n, a, k) : (n, a)^{k+1} = (n, a)^{k+1}\) 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

\[
(\text{colim}(A, a) \to E) = \left(\sum_{e: \Sigma_{A} \to E} \Pi_{x: \Sigma_{A}} e(x) = e(x)\right)
\]

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

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

1. We get a function \(\text{colim}(\tau, H)\) or \(\tau_{\infty} : A_{\infty} \to A'_{\infty}\).

2. The sequential colimit is \(1\)-functorial. This means the following three things: If \((\sigma, K) : (A', a') \to (A'', a'')\) 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 \((\tau', H') : (A, a) \to (A', a')\) and \(q : \Pi_{n:N} \tau(n) \sim \tau'(n)\) and we can fill the following square for all \(a : A(n)\)

\[
\begin{array}{ccc}
\tau(n, a) & \sim & \tau'(n, a) \\
\text{ap}(\tau(n, a)) & \sim & \text{ap}(\tau'(n, a))
\end{array}
\]
We now shift our attention to a dependent version of type which can be visualized as the family of type sequences to equifibered type sequences over $\tau(n + 1, a, x) \xrightarrow{\pi_{n+1}(H_a(n))} \tau'(n + 1, a(n, a))$

\[ a'(n, \tau(n, a)) \xrightarrow{\text{ap}_{e(n, a)}(\pi(n, a))} a'(n, \tau'(n, a)) \]

then $\tau_{\omega} \sim \tau_{\omega}$. 3. If $\tau$ is a natural equivalence then $\tau_{\omega}$ is an equivalence.

**Proof:**

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

\[ \text{ap}_{\tau_\omega}(\pi(n, a)) := \text{ap}_{i(n+1)}(H(a)) \cdot \pi(n, \tau(n, a)) \]

: $i(n + 1, \tau(n + 1, a(n, a))) = i(n, \tau(n, a))$. 2. All three parts are by straightforward induction on $A_{\lambda_\omega}$.

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

\[ \tau_{\omega} \circ \tau_{\omega} \sim (\tau^{-1}_{\omega} \circ \tau_{\omega}) \sim 1_{\omega} \sim \text{id}_{\lambda_\omega} \]

\[ \square \]

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 $(\tau, b) : \text{FibSeq}(A, a)$ with

\[ B : \Sigma A \rightarrow U \]

\[ b : \Pi_{x : x \equiv A} B(x) \rightarrow B(x^+) \]

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

\[ B(x) \xrightarrow{b(x)} B(x^+) \xrightarrow{b(x^+)} B(x^{++}) \xrightarrow{b(x^{++})} \cdots \]

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

\[ \text{colim}(A, a) \rightarrow U \]

\[ = \Sigma_{C : C(A, a) \rightarrow U} \Pi_{x : x \equiv A(n)} C(n + 1, a(n, a)) \]

\[ \simeq C(n, a) \]

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:

\[ i(n, (a, b)) \mapsto i(n, a) \]

\[ \kappa(n, (a, b)) \mapsto \kappa(n, a) \]

We can informally describe the type $\text{colim}(\Sigma_{(A, a)}(B, b))$ as a "colim after $\Sigma$". The natural questions now are, what does the dual type "$\Sigma$ 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 $B_{\omega}$.

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

\[ C_{\omega}(n, a) := \text{colim}(k \mapsto B((a, n)^{+k}), k \mapsto b((a, n)^{+})^{+}) \]

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

\[ C_{\omega}(n + 1, a(n, a)) \]

\[ \simeq \text{colim}(k \mapsto B((a, n)^{+k}), k \mapsto b((a, n)^{+k})) \]

\[ \simeq \text{colim}(k \mapsto B((a, n)^{+k+1}), k \mapsto b((a, n)^{+k+1})) \]

\[ \simeq \text{colim}(k \mapsto B((a, n)^{+k}), k \mapsto b((a, n)^{+})^{+}) \]

\[ \simeq C_{\omega}(n, a) \]

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

\[ Q(n, a, k) := \text{ap}_b(\text{ap}_k(q(n, a, k))) \]

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 $\delta(n, a, k)$.

\[ \text{B}((a, n)^{+k}) \xrightarrow{\text{Q}(n, a, k)} \text{B}((a, n)^{+k+1}) \]

\[ \text{B}((a, n)^{+k+1}) \xrightarrow{\text{Q}(n, a, k + 1)} \text{B}((a, n)^{+k+2}) \]

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

\[ \text{B}((a, n)^{+k}) \xrightarrow{\text{ap}_b(\text{ap}_k(q(n, a, k)))} \text{B}((a, n)^{+k+1}) \]

\[ \text{B}((a, n)^{+k+1}) \xrightarrow{\text{ap}_b(\text{ap}_k(q(n, a, k)))} \text{B}((a, n)^{+k+2}) \]
We can fill this diagram by generalizing the path $q(n, a, k): (n, a)^{(k + 1)} = (n, a)^{(k + 1)}$ together with its endpoints and doing path induction. This gives the map $e_1(n, a)$. Moreover, $(Q, \delta)$ 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:

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

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

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

- $B_\infty(e(n, a)) \equiv C_\infty(n, a)$
- $\text{ap}_n(B_\infty(n, a)) = c(n, a)$

5 Colimits and Sums

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

\[
\begin{align*}
(i(n + 1, a(n, a)), i(0, b(n, a, b))) & \quad \text{and} \quad (i(n, a), i(0, b))
\end{align*}
\]

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

\[
\begin{align*}
(i(n + 1, a(n, a)), i(0, b(n, a, b))) & \quad \text{and} \quad (i(n, a), i(1, b(n, a, b))) \\
\end{align*}
\]

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

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

we have a path $\Delta(\alpha, F_\ast, y) : (x_1, y) = (x_2, F(y))$ defined by induction on $\alpha$ and $F_\ast$. 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 $\Sigma$).** We have:

\[
\text{colim}(\Sigma_{A_\infty}(B, b)) = \sum_{A_\infty}B_\infty(x)
\]

as witnessed by the equivalence $F$ below:

\[
i(n, (a, b)) \mapsto (i(n, a), i(0, b))
\]

\[
\kappa(n, (a, b)) \mapsto \Delta(\kappa(n, a), c_\infty(n, a), i(0, b(n, a, b))) \cdot \text{ap}_n(i(n, a), i(0, b(n, a, b))
\]

that commutes with the canonical projections:

\[
\begin{align*}
\text{colim}(\Sigma_{A_\infty}(B, b)) & \quad F \\
\sum_{A_\infty}B_\infty(x) & \quad \pi_1&
\end{align*}
\]

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 $i(n, (a, b))$ the maps agree definitionally. To show that $\kappa(n, a) = \text{ap}_n(\pi_1(F_\ast(x(n, a))))$, 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 $y_2 : B_\infty(x_2)$
- $F : B_\infty(x_1) \rightarrow B_\infty(x_2)$
- $F_\ast : \text{ap}_n(B_\infty(a)) = F$
- $\beta : F(y_1) = y_2$

we have $\alpha = \text{ap}_n(\Delta(\alpha, F_\ast, y_1) \cdot \text{ap}_n(y_2, y_1))$. \hfill $\square$

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, path-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 $\sum A_\infty B_\infty(x)$. 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 $\sum A_\infty B_\infty(x)$ as if we were constructing a map out of $\text{colim}(\Sigma_{A_\infty}(B, b))$, using $F$ to appropriately mediate between the two types:

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

*Given*

- $E : (\sum_{A_\infty}B_\infty(x)) \rightarrow U$
- $e : \prod_{n \geq 0} \prod_{i \in (n, a)} \prod_{k \in (n, a, b)} E(F(i(n, a), b))$
The proof of our main result, Theorem 5.1, is now easy:

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

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

\[ \begin{align*}
&\exists \mathcal{F} : \Pi_{\Sigma x : A(n)} \mathcal{B}(x) \\
&\mathcal{F}(a, b) = (n, a) \quad \text{such that for any } n : \mathbb{N}, a : A(n), b : \mathcal{B}(n, a), \mathcal{F}(a, b) = (n, a, b)
\end{align*} \]

there is a function \( G : \Pi_{X : \Sigma x : A(n)} \mathcal{B}(x) \rightarrow E \) such that for any \( n : \mathbb{N}, \) \( a : A(n) \), \( b : \mathcal{B}(n, a) \)

\[ G(a, b) = \mathcal{F}(a, b) \]

\[ \mathcal{F} \text{ is epic.} \]

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

Proof of equivalence. We construct the inverse \( F^{-1} \) by recursion on \( \Sigma x : A(n) \mathcal{B}(x) \) as in Lemma 5.3:

\[ F(a, b) = (n, a, b) \]

\[ \mathcal{F}(a, b) = \mathcal{F}(a, b) \]

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

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 curved version of \( G \), the map \( x, y : \Pi \mathcal{G}(x, y) : \Pi_{X : \Sigma x : A(n)} \mathcal{B}(x, y) \) by nested sequential colimit induction. For the outer induction on \( A(n) \), we need to map \( (n, a) \) to a function

\[ \mathcal{F}(a, b) \]

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

\[ \kappa(n, a) \#_{(x \mapsto \Pi_{\mathcal{B}(x)} \mathcal{G}(x, y))} \mathcal{G}(n + 1, a(n, a)) = \mathcal{G}(n, a) \quad (1) \]

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

\[ h(a, k, b) : E(a, (n, a), \theta(k, b)) \]

\[ H(n, a, k, b) = \kappa(k, b) \#_{E(a, (n, a), \theta(k, b))} \]

\( h(n, a, k + 1, b((n, a)^k, b)) = h(n, a, k, b) \)

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 \( C_n(a, n) \) and the latter from \( C_{n+1}(a, n) \). But these sequential colimits are equivalent via \( c(a, n) \), so relating the maps \( g(n + 1, a(a, n)) \) and \( g(n, a) \) comes down to relating \( g(n + 1, a(n, a)) \) and \( g(n, a, c(a, n)) \) for any argument \( b : C_n(a, n) \). The respective values lie in different fibers of \( E \), namely over \( (n + 1, a(n, a)), b \) versus over \( (n, a), c(n, a), b \). But this is fine: the path

\[ \Delta(\kappa(n, a), c_*(n, a), b) \]

equates the pairs.

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

\[ \Pi_{X : \Sigma x : A(n) \mathcal{B}(x)} \mathcal{G}(n, a, c(n, a)) = \Delta(\kappa(n, a), c_*(n, a), y) \#_{E} \mathcal{G}(n + 1, a(n, a), y) \]

(2)

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

\[ \begin{align*}
&X : \Sigma x : A(n) \mathcal{B}(x) \\
&X(1) \quad : \mathcal{B}(n) \\
&F : \mathcal{B}(n) \rightarrow \mathcal{B}(n) \\
&F : \mathcal{B}(n) \rightarrow F(n) \\
&f_1 : \Pi_{X : \Sigma x : A(n) \mathcal{B}(x)} \mathcal{G}(x, y) \\
&f_2 : \Pi_{X : \Sigma x : A(n) \mathcal{B}(x)} \mathcal{G}(x, y)
\end{align*} \]

we have an equivalence \( \mathcal{I}(\alpha, F_*, f_1, f_2) : \)

\[ \alpha \#_{(x \mapsto \Pi_{\mathcal{B}(x)} \mathcal{G}(x, y))} \mathcal{G}(x, y) = \mathcal{G}(x, y) \]

\[ \mathcal{I}(\kappa(n, a), c_*(n, a), y) \#_{E} \mathcal{G}(n + 1, a(n, a), y) \]

(2)

We define \( I \) by induction on \( \alpha \) and \( F_* \). This reduces the above equivalence to \( (f_1 = f_2) \sim (f_1 \sim f_2) \), witnessed by the map \( \text{happ}(f_1, f_2) \).

It thus suffices to produce a witness \( \omega(n, a) \) for (2), which will give us the desired witness for (1), explicitly described
below:
\[
I\left(\kappa(n, a), c_\star(n, a), g(n+1, a(n, a)), g(n, a)\right)^{-1} \omega(n, a)
\]

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

\[
\mu(n, a, k, b) : g(n, a, c(n, a), i(k, b)) = \\
\Delta(\kappa(n, a), c_\star(n, a), i(k, b)) \#_E g(n+1, a(n, a), i(k, b))
\]

and an equality:

\[
\kappa(k, b) \#_{y \rightarrow g(n, a, c(n, a), y)} = \Delta(\kappa(n, a), c_\star(n, a), y) \#_E g(n+1, a(n, a), y)
\]

\[
\mu(n, a, k + 1, b((n, a)^{++k})) = \mu(n, a, k, b)
\]

(3)

We can now divide the proof into the following steps:

- The point-point case: defining \(h\).
- The path-point case: defining \(\mu\).
- 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 \(\mu\). By definition of \(g\) we have

\[
g(n + 1, a(n, a), i(k, b)) \equiv h(n + 1, a(n, a), k, b)
\]

and similarly

\[
g(n, a, c(n, a), i(k, b)) \equiv g(n, a, i(k + 1, Q(n, a, k, b))) = h(n, a, k + 1, Q(n, a, k, b))
\]

hence we need

\[
\mu : h(n, a, k + 1, Q(n, a, k, b)) = \\
\Delta(\kappa(n, a), c_\star(n, a), i(k, b)) \#_E h(n + 1, a(n, a), k, b)
\]

We recall that the map \(Q(n, a, k) := ap_{\beta}(q(n, a, k))\) 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 \(k, n, a, b \mapsto h(n, a, k, b) : \Pi_{k: \mathbb{N}} \Pi_{n: \mathbb{N}} \Pi_{a: A(n)} \Pi_{b: B(n(a)^{++})} E(i(n, a), i(k, b))\) by induction on \(k\). In the zero case, we need a point in the fiber \(E(i(n, a), i(0, b))\) and this is just the assumption \(e(n, a, b) : E(F(i(n, a), b)))\). 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 \(\mu\), but we cannot do so directly because \(b\) does not have the form \(b := Q(n, a, k)\) 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

\[
A, B : U \text{ and } a : A = B
\]

\[
P : B \rightarrow U
\]

\[
f : \Pi_{a: A} P(\alpha(a))
\]

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

\[
\beta(\alpha, P, f) : \Pi_{a: A} J(\alpha, P, f, \alpha(a)) = f(a)
\]

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 \(\mu\):

\[
h(n, a, 0, b) := e(n, a, b)
\]

\[
h(n, a, k + 1) := J(\alpha P_{\beta}(q(n, a, k)),
\]

\[
b : B((n, a)^{++k}) \mapsto E(i(n, a), i(k + 1, b)),
\]

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

\[
\Delta(\kappa(n, a), c_\star(n, a), i(k, b)) \#_E h(n + 1, a(n, a), k, b)
\]

\[
\mu(n, a, k) := \beta f(\alpha P_{\beta}(q(n, a, k)))
\]

\[
b : B((n, a)^{++k + 1}) \mapsto E(i(n, a), i(k + 1, b)),
\]

\[
b : B((n, a)^{++k + 1}) \mapsto
\]

\[
\Delta(\kappa(n, a), c_\star(n, a), i(k, b)) \#_E h(n + 1, a(n, a), k, b)
\]

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 \(i(k + 1, b((n, a)^{++k}))\) and \(i(k, b)\) by something more general:

\[
y_1, y_2 : C_w(n + 1, a(n, a)) \text{ and } \beta : y_1 = y_2
\]

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

\[
u_1 : g(n, a, c(n, a), y_1) = \\
\Delta(\kappa(n, a), c_\star(n, a), y_1) \#_E g(n + 1, a(n, a), y_1)
\]

\[
u_2 : g(n, a, c(n, a), y_2) = \\
\Delta(\kappa(n, a), c_\star(n, a), y_2) \#_E g(n + 1, a(n, a), y_2)
\]

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

\[
ap_{\kappa(n, a)}(k, b) = ap_{\kappa(n, a, b)}(\delta(n, a, k, b)) \cdot \kappa(k + 1, Q(n, a, k, b))
\]

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

\[
z : C_w(n, a)
\]

\[
y_1 : c(n, a) y_1 = z
\]

\[
y_2 : z = c(n, a) y_2
\]

\[
\theta : ap_{\kappa(n, a)}(\beta) = y_1 \cdot y_2
\]

In this abstract scenario, we want to relate \(u_1\) and \(u_2\) over the path \(\beta\). Since \(u_1\) is a path in \(E(i(n, a), c(n, a), y_1)\) and \(u_2\)
is a path in $E(i(n, a), c(n, a) y_2)$, 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 $E(i(n, a), -)$
over $ap_{E(n, a)}(\beta)$, which we could use to carry $u_1$ over. But
since we know $ap_{E(n, a)}(\beta) = y_1 \cdot y_2$, it is better to use
the transports over $y_1$ and then $y_2$ 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 $\epsilon$, can be seen as witnessing the naturality
of $\Delta(k(n, a), c_*(n, a))$ 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 $\beta$.

Putting all this together, we have the following generalization:

ey_1, y_2 : C_{\omega}(n + 1, a(n, a)) and $\beta : y_1 = y_2$

there is a path $\epsilon$ as indicated in figure 2 such that for any

$u_1 : g(n, a, c(n, a) y_1) = \Delta(k(n, a), c_*(n, a), y_1) \eta_E g(n + 1, a(n, a), y_1)$

$u_2 : g(n, a, c(n, a) y_2) = \Delta(k(n, a), c_*(n, a), y_2) \eta_E g(n + 1, a(n, a), y_2)$

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

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

After carrying out this procedure, the two verticals of $\epsilon$ both
reduce to

$\Delta(k(n, a), c_*(n, a), y_1) \eta_E g(n + 1, a(n, a), y_1)$

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
$\text{refl} = \text{refl} \cdot \text{id}_{a(n, a)} \cdot \text{refl} \cdot \text{refl} \cdot u_2^{-1}$ and this is clearly
the case. We will refer to the edge we just constructed as
$\epsilon(y_1, y_2, \theta)$. Our intended instantiation is thus:

- $y_1 := i(k + 1, b((n, a)^{1+k}) b)$
- $y_2 := i(k, b)$
- $\beta := k(b, b)$
- $z := i(k + 2, b((n, a)^{k+1}) Q(n, a, k, b))$
- $y_1 := ap_{i(k+2)}(\delta(n, a, k, b))$
- $y_2 := k(k + 1, Q(n, a, k, b))$
- $\theta := \beta(n, a, k, b)$
- $u_1 := \mu(n, a, k + 1, b((n, a)^{1+k}) b)$
- $u_2 := \mu(n, a, k, b)$

So to prove it suffices to establish the equality in figure 4. On
the left hand side, the path $ap_{E(n, a)}(\kappa(k + 1, Q(n, a, k, b)))$
is equal to $H(n, a, k + 1, Q(n, a, k, b))$. On the right hand side, the
path $ap_{E(n+1, a(n, a))}(\kappa(k))$ 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 re-
call that given a sequence $(A, a)$ we have a function $(-)^{\ast k} : A(n) \to A(n + k)$. Sometimes the codomain we need is not
exactly of the form $A(n + k)$. For example, if $a : A(0)$, then
$(0, a)^{\ast k}$ has type $A(0 + k)$ which is not definitionally equal
to $A(k)$. So, generally, given a proof that $n \leq m$ we define
$a^{n \leq m} : A(n) \to A(m)$ inductively by taking $a^{n \leq 0} := \text{id}_{A_n}$
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
$a^{n \leq m + k}(a) = (n, a)^{1+k}$.

The following lemma states that $(0)$ 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.
We also recall the following lemma:

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

We also recall the following lemma:

**Lemma 7.2.** Let \(a : A\) and let \(B : A \to U\) be a type family with \(b : B(a)\). Then the following are equivalent:

1. The canonical family of maps \(\Pi x. A(x = x) \to B(x)\) 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

\[
(i(n, a) =_{A_\infty} i(n, a')) \simeq \colim A((n, a) + k) = (n, a) + k)
\]

**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 \(B(n, a) \defeq \delta^\leq n(a_0) = A(n) a\) and \(b(n, a, p) \defeq \ap a_0(p)\). Now we use Theorem 5.1 to see that the total space of \(B_\infty\) is contractible.

\[
\Sigma_{a : A_\infty} B_\infty(a) = \colim A((n, a)^+ k (a_0) = a) \simeq \colim A(1) \simeq 1.
\]
Sequential Colimits in Homotopy Type Theory

Since $i(0, \text{refl}_a) : B_\omega(i(0, a_0))$ we can now conclude by Lemma 7.2 that

$$i(0, a_0) = A_\omega(i(0, a_0))$$

$$\simeq B_\omega(i(0, a_0))$$

$$\equiv \colim((0, a_0)^{n \leq 0 + k} = A_{(0 + k)}(0, a)^{n + k})$$

$$\equiv \colim((0, a_0)^{n + k} = A_{(0 + k)}(0, a)^{n + k}).$$

The last step uses that $a_0^{n + k}((0, a_0)) = (0, a_0)^{n + k}$. For general $n$, we denote the sequence $(k \mapsto A_n(a + k), k \mapsto a_n + k)$ as $S_n(a, a_0)$ or $(S_n(A), S_n(a))$. By iterating Lemma 3.5 we get an equivalence $e : A_\omega \simeq (S_n(A), a)_\omega$. For $a, a' : A(n)$ we can now compute:

$$(i(n, a) = A_\omega(i(n, a')))$$

$$\simeq (e_n(i(n, a)) = \colim(S_n(A, a)) e_n(i(n, a')))$$

$$\simeq (i(0, a) = \colim(S_n(A, a)) i(0, a'))$$

$$\simeq \colim((0, a)^{n + k} = A_{(0 + k)}(0, a')^{n + k})$$

$$\simeq \colim((n, a)^{n + k} = A_{(n + k)}(0, a')^{n + k}).$$

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

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

A pointed type sequence is a type sequence where all types and maps are pointed. We will denote the basepoint of $A_\omega$ by $a_0$, and now the sequential colimit is pointed with basepoint $i(0, a_0)$. The next corollary shows that colimits commute with the 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

$$\Omega A_\omega \simeq \colim(n \mapsto \Omega A(n), n \mapsto \Omega a(n)).$$

**Proof.** We compute

$$\Omega A_\omega \simeq \colim_n((0, a_0)^{n \leq 0 + n} = (0, a)^{n + n})$$

$$\simeq \colim_n(a_0^{0 \leq n}(0, a_0) = a^{0 \leq n}(0, a))$$

$$\simeq \colim_n((0, a_0) \mapsto \Omega a(n), n \mapsto \Omega a(n)).$$

The first equivalence is Theorem 7.3, and the last two are by functoriality of the colimit. The second equivalence uses that $(0 + n, (0, a)^{n + n}) = (n, a^{0 \leq n}(a_0))$ 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.

Given a natural transformation $\tau : (A', a') \to (A, a)$ we can define the homotopy fiber of $\tau$ 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 $\tau : (A', a') \to (A, a)$ and a point $x : A_\omega$. Then

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

**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.

$$\Sigma_{x : A_\omega}(\text{fib}_\tau(\omega)(x)) \simeq \colim_k(\Sigma_{x : A(n)}(\text{fib}_\tau(n)(x))) \simeq A_\omega'$$

$$\xymatrix{ \Sigma_{x : A_\omega}(\text{fib}_\tau(\omega)(x)) \ar[r]^{\pi_1} \ar[d]_{\tau_\omega} & \colim_k(\Sigma_{x : A(n)}(\text{fib}_\tau(n)(x))) \ar[d]^{\pi_k} \ar[r]^{\tau_\omega} & A_\omega' \ar[d]_{\tau_\omega} \ar[l]^{\pi_k} \\ \xymatrix{ A_\omega \ar[r]^{\tau_\omega} & A_\omega'} }$$

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,

$$\text{fib}_{\tau\omega}(x) \simeq \text{fib}_{\pi_1}(x) \simeq (\text{fib}_\tau(\omega))(x).$$

The following corollary specifies the interaction of colimits and truncations, truncadedness 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 homotopy 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 $A_\omega$ is $k$-truncated.
2. We have an equivalence

$$\|A_\omega\|_k \simeq \colim_n(\|A(n)\|_k, \|a(n)\|_k).$$

3. If $A(n)$ is $k$-connected for all $n : \mathbb{N}$, then $A_\omega$ is $k$-connected.
4. Given a natural transformation $\tau : (A', a') \to (A', a')$ such that $\tau(n)$ is $k$-truncated (k-connected) for all $n$, then $\tau_\omega$ is $k$-truncated (k-connected).
5. If $A(n)$ is $k$-truncated (k-connected) for all $n$, then $\tau(0)$ is $k$-truncated (k-connected).

By 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 $(m)$ 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 \( A_{\infty} \cong A(0) \)
by Lemma 7.1, hence \( A_{\infty} \) 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 \( \iota(n, a) = \iota(n + m, a_n \equiv n + m(a(i), \iota(m, a')) \),
therefore the type \( \iota(n, a) = \iota(m, a') \) is equivalent to
\[ \iota(n + m, a_n \equiv n + m(a(i), \iota(m, a'))). \]
Therefore it suffices to show that the latter equality
type is \( k' \)-truncated. By Theorem 7.3 we need to show that the
type
\[
colim_{\ell}(n + m, a_n \equiv n + m(a(i), \iota(m, a')) + \ell)
\]
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 \( A_{\infty} \rightarrow \colim_{\ell}(\|A(n)\|_k, \|a(n)\|_k) \). Because the right hand side is \( k \)-truncated, this induces a map
\[ g : \|A_{\infty}\|_k \rightarrow \colim_{\ell}(\|A(n)\|_k, \|a(n)\|_k). \]
For the other direction, we define a map
\[ h : \colim_{\ell}(\|A(n)\|_k, \|a(n)\|_k) \rightarrow \|A_{\infty}\|_k \]
by
\[ \iota(n, \|a\|_k) \mapsto \iota(n, a) \]
\[ \kappa(n, \|a\|_k) \mapsto \kappa(n, a) \]
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 \( \|A_{\infty}\|_k \cong \colim_{\ell}(\|A(n)\|_k) \) is contractible.
4. Let \( x : A_{\infty} \). We need to show that \( \text{fib}_{r_x}(x) \) 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
\[ \text{fib}_{r_x}(\iota(n, a)) \cong (\text{fib}_{r_x}(\iota(n, a))). \]
by Corollary 7.5. Since \( \text{fib}_{r_x}(x) \) is \( k \)-truncated (\( k \)-
connected) for all \( n \) and \( x \), we know that its colimit
\( \text{fib}_{r_x}(\iota(n, a)) \) is \( k \)-truncated (\( k \)-connected), by part 1
or 3. This finished the proof.
5. Consider the natural transformation
\[
\begin{array}{ccccccc}
A(0) & \longrightarrow & A(0) & \longrightarrow & \ldots & \longrightarrow & \colim_{\ell}(A(0)) \\
\downarrow^{\alpha_{0 \leq 0}} & & \downarrow^{\alpha_{0 \leq 1}} & & \ldots & & \downarrow \\
A(0) & \longrightarrow & A(1) & \longrightarrow & \ldots & \longrightarrow & A_{\infty}
\end{array}
\]
The maps \( \alpha_{0 \leq n} : A(0) \rightarrow A(n) \) are \( k \)-truncated (\( k \)-
connected) and form a natural transformation. Therefore,
by part 4 the map \( \alpha_{0 \leq \infty} : \colim_{\ell}(A(0)) \rightarrow A_{\infty} \)
is \( k \)-truncated (\( k \)-connected). The fiber of \( \iota(0) \) over \( x \in A_{\infty} \)
is the same as the fiber of \( a_{0 \leq \infty} \) over \( x \), and therefore
\[ \iota(0) \] is \( k \)-truncated (\( k \)-connected). \( \Box \)

8 Conclusion
We presented here the original proof of the conjecture that in homotopy type theory sequential colimits commute with \( \Sigma \)-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, Boucquet’s independent proof [Boucquet 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 \( n \) : \( N \) but only for the base case \( n = 0 \)), plus an additional constructor lift that takes an element \( x : \colim(n \mapsto A(n + 1), n \mapsto a(n + 1)) \) 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 \( B_{\infty} \) 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 \( \text{lift}(f) \), 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 Boucquet 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 typal computation rules corresponds to a simple rewrite).

We have used the proof assistant Lean 2\(^1\) to formalize our results. Lean 2 is an older version of Lean\(^2\) 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/
Sequential Colimits in Homotopy Type Theory

References


B. Ahrens, P. Capriotti, and R. Spadotti. 2015. Non-Wellfounded Trees in
Homotopy Type Theory. In Typed Lambda Calculi and Applications
17–30.

Mini-Workshop (Oberwolfach): The Homotopy Interpretation of Con-


Types. Mathematical Proceedings of the Cambridge Philosophical Society
146, 1 (2009), 45–55.

R. Böcquet. 2017. Proof of the Commutativity of Sums and Sequential

https://github.com/RafaelBocquet/CoinductiveSequentialColimits.

G. Brunerie. 2016. On the Homotopy Groups of Spheres in Homotopy Type

R. Graham. 2017. Synthetic Homology in Homotopy Type Theory. Master’s

J. Christensen, M. Opie, E. Rijke, and L. Scoccola. 2018. Localization in

C. Cohen, T. Coquand, S. Huber, and A. Mörtberg. 2016. Cubical Type

D. Licata and G. Brunerie. 2015. A Cubical Approach to Synthetic Homotopy

(2011). Available at https://homotopytypetheory.org/2011/04/24/higher-
inductive-types-a-tour-of-the-menagerie/.

P. LeFanu Lumsdaine and M. Shulman. 2012. The Simplicial Model of Univa-

N. Kraus and J. von Raumer. 2019. Path Spaces of Higher Inductive Types
in Homotopy Type Theory. In To appear in Logic in Computer Science
(LICS 2019).

D. Licata and G. Brunerie. 2015. A Cubical Approach to Synthetic Homotopy

(2011). Available at https://homotopytypetheory.org/2011/04/24/higher-
inductive-types-a-tour-of-the-menagerie/.

P. LeFanu Lumsdaine and M. Shulman. 2017. Semantics of Higher Inductive


ph.utexas.edu/category/2011/04/homotopy_type_theory_vi.html.

Post on the Homotopy Type Theory blog.

homotopytypetheory.org/2013/08/08/spectral-sequences/.

M. Shulman. 2019. All (∞,1)-Toposes Have Strict Univalent Universes.


The Univalent Foundations Program, Institute for Advanced Study. 2013.
Homotopy Type Theory - Univalent Foundations of Mathematics. Univalent
Foundations Project.

F. van Doorn. 2016. Constructing the Propositional Truncation Using Non-

V. Voevodsky. 2010. The Equivalence Axiom and Univalent Models of Type

talk at the Workshop on Logic, Language, Information and Computation
(WoLLIC 2011).