Coslice Colimits in Homotopy Type Theory

Perry Hart
Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, USA

Kuen-Bang Hou (Favonia)
Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, USA

Abstract

We contribute to the theory of (homotopy) colimits inside homotopy type theory. The heart of our work characterizes the connection between colimits in coslices of a universe, called *coslice colimits*, and colimits in the universe (i.e., ordinary colimits). To derive this characterization, we find an explicit construction of colimits in coslices that is tailored to reveal the connection. We use the construction to derive properties of colimits. Notably, we prove that the forgetful functor from a coslice creates colimits over trees. We also use the construction to examine how colimits interact with orthogonal factorization systems and with cohomology theories. As a consequence of their interaction with orthogonal factorization systems, all pointed colimits (special kinds of coslice colimits) preserve n-connectedness, which implies that higher groups are closed under colimits on directed graphs. We have formalized our main construction of the coslice colimit functor in Agda.

2012 ACM Subject Classification Theory of computation → Type theory

Keywords and phrases colimits, homotopy type theory, category theory, higher inductive types, synthetic homotopy theory

Digital Object Identifier 10.4230/LIPIcs.CSL.2025.46

Related Version *Technical report*: `https://doi.org/10.48550/arXiv.2411.15103` [6]

Supplementary Material *Software (Agda Code)*: `https://github.com/PHart3/colimits-agda/tree/v0.1.0` [7]

Funding This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-21-1-0009. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force.

Acknowledgements We thank the anonymous reviewers for their feedback that improved the writing of this paper. We also thank the anonymous reviewer for HoTT/UF 2023 who pointed out the relationship between adjunctions and factorization systems.

1 Introduction

Homotopy type theory (HoTT) extends Martin-Löf type theory (MLTT) with univalence and higher inductive types [27]. The key feature of HoTT is that all types behave as homotopy types of topological spaces [9]. Thus, with HoTT, we can use purely type-theoretic methods to prove new properties of spaces. Moreover, higher inductive types (HITs) let us bring a vast range of spaces into HoTT. As a result, HoTT is a useful system for developing synthetic homotopy theory and formalizing it in proof assistants like Coq and Agda [5, 8].

We study HITs arising as *(homotopy) colimits* in coslices of a universe, called *coslice colimits*. Coslices of a universe are type-theoretic versions of coslice categories. A colimit in a category is an object formed by gluing together simpler objects in a coherent fashion. The *coherent* requirement ensures that the colimit has a universal property, which reduces proofs about the colimit to proofs about the simpler objects it is built out of. When these objects are spaces, perhaps endowed with extra structure, colimits built out of them find

33rd EACSL Annual Conference on Computer Science Logic (CSL 2025).
Editors: Jörg Endrullis and Sylvain Schmitz; Article No. 46; pp. 46:1–46:20
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

wide use in homotopy theory. For example, the class of HITs we study includes colimits of *pointed spaces*. Such colimits are key to the *Brown representability theorem*, which is about homotopy functors on the (∞-)category of pointed connected spaces. Indeed, its proof relies on the fact that this category is generated under colimits by compact cogroups.

1.1 Contributions

In this section, we explain the contributions of the paper along with its organization. We start by outlining the heart of the paper, which we call *the main connection*. Afterward, we describe three independent applications of the main connection in synthetic homotopy theory. Details, proofs, and related additional results are found in our associated technical report [6]. Further, we have formalized in Agda our construction of A-colimits as well as the universality of ordinary colimits for Corollary 21.

1.1.1 The main connection (Section 5)

Suppose $\mathcal{U}$ is a universe and A is a type in $\mathcal{U}$. We want to construct colimits in the *coslice* $A/\mathcal{U}$, or A-*colimits*. The (wild) category $A/\mathcal{U}$ has objects $\sum_{T:\mathcal{U}} A \to T$ and morphisms $X \to_A Y :\equiv \sum_{k:\mathsf{pr}_1(X)\to\mathsf{pr}_1(Y)} k \circ \mathsf{pr}_2(X) \sim \mathsf{pr}_2(Y)$. Here, $\sim$ is defined by $f_1 \sim f_2 :\equiv \prod_{x:X} f_1(x) = f_2(x)$ for any dependent functions $f_1, f_2 : \prod_{x:X} Y(x)$, called the type of *homotopies* from f_1 to f_2.

HoTT has a general schema for HITs that would let us simply postulate A-colimits. We, however, explicitly construct A-colimits with just the machinery of MLTT augmented with pushouts (Section 5.3).[1] We take this different approach to reveal the connection between A-colimits and their underlying colimits in $\mathcal{U}$. In fact, our construction is *not* a case of a general method to encode higher-dimensional HITs with pushouts but rather tailored to reveal this connection.

Why do we care about this connection? It sheds light on three established areas of synthetic homotopy theory. We preview them now and return to them in Sections 6–8.

The universality of colimits (Section 6)

The *universality* of colimits is a special feature of locally cartesian closed (LCC) ∞-categories, such as that of spaces. The main connection will establish a well-known classical result inside type theory: The forgetful functor $A/\mathcal{U} \to \mathcal{U}$ *creates* colimits of diagrams over contractible graphs (Theorem 18), which make up a large subclass of graphs.[2] Examples of such colimits include sequential colimits [25]. With the forgetful functor creating colimits, we can transfer universality of ordinary colimits to A-colimits over contractible graphs (Corollary 21). This is notable as LCC ∞-categories are not closed under coslices.

The categories of higher groups are cocomplete (Section 7)

A striking feature of colimits is their interaction with (orthogonal) factorization systems. In Section 7, we use the main connection to show that colimits in $A/\mathcal{U}$ preserve left classes of maps of factorization systems on $\mathcal{U}$. It is significant that we consider factorization systems on $\mathcal{U}$ rather than $A/\mathcal{U}$. We could derive a similar preservation theorem for systems on $A/\mathcal{U}$ directly from the universal property of an A-colimit. In practice, however, the factorization

[1] A theoretical advantage of such a construction is that pushouts, the simplest nontrivial HITs, can be postulated with a less powerful schema than that required to postulate A-colimits.

[2] For a definition of *creating (co)limits*, see [16].

systems we tend to care about are on $\mathcal{U}$. Since the main connection relates the action of A-colimits on maps to the action of the underlying ordinary colimits on maps (Section 5.4), we manage to deduce the preservation theorem for systems on $\mathcal{U}$.

To prove this theorem, we find it useful to develop the theory of factorization systems in a more general setting than $\mathcal{U}$. In Section 4.1, we study such systems on *wild categories*, which make up one approach to category theory in HoTT. We prove that if a functor F of well-behaved wild categories with factorization systems has a right adjoint G, then F preserves the left class when G preserves the right class (Theorem 13). We combine this result with the main connection to deduce the desired preservation property.

When we focus on the (n-connected, n-truncated) factorization system on $\mathcal{U}$ [27, Chapter 7.6] and take A as the unit type, the main connection shows that the colimit of every diagram of pointed n-connected types is n-connected. One useful corollary of this is that the higher category $(n, k)\,\mathsf{GType}$ of k-tuply groupal n-groupoids considered by [2] is cocomplete on (directed) graphs for all truncation levels $-2 \leq n \leq \infty$ and $-1 \leq k < \infty$ (Example 23).

Cohomology sends colimits to weak limits (Section 8)

Finally, we examine how colimits interact with cohomology theories, which are important algebraic invariants of spaces. To do so, we consider *weak limits*, which are key ingredients in the Brown representability theorem (BRT). A weak colimit in a category need not satify the uniqueness property required of a colimit. The BRT specifies conditions for a presheaf on the homotopy category $\mathbf{Ho}(\mathbf{Top}_{*,c})$ of pointed connected spaces to be representable. The standard proof of the BRT requires the presheaf to send countable homotopy colimits in $\mathbf{Top}_{*,c}$ to weak limits in **Set** [14, Section 1.4.1]. Eilenberg-Steenrod cohomology theories enjoy this property as set-valued functors.

In Section 8, we use the main connection to establish a restricted, type-theoretic version of this property. From the main connection we derive another construction of A-colimits, as pushouts of coproducts (Corollary 26), which mirrors a well-known classical lemma. We take A as the unit type and combine the new construction with the Mayer-Vietoris sequence to find that cohomology takes finite colimits to weak limits assuming the axiom of choice.

2 Additional related work

2.1 Construction of nonrecursive 2-HITs

The HITs we consider are nonrecursive 2-HITs, in the sense that they have only nonrecursive constructors of points and of paths of dimension one or two. Van Doorn et al. explicitly construct nonrecursive 2-HITs in MLTT augmented with pushouts [28, Section 5]. When specialized to A-colimits, however, their construction has a significantly different form from ours and does not directly lead to the properties of A-colimits we derive. Moreover, they do not prove the full induction principle enjoyed by the 2-HIT for their construction, whereas we do for ours. The full induction principle is necessary (and sufficient) to characterize the 2-HIT uniquely.

2.2 Orthogonal factorization systems

Our work also builds on the theory of factorization systems. Such systems play important roles in model category theory [20], a key framework for classical homotopy theory. Moreover, in type theory, Rijke et al. have shown that factorization systems on $\mathcal{U}$ are closely connected to modalities [22], which are important in logic. We extend factorization systems to wild categories other than $\mathcal{U}$. Moreover, we lift factorization systems on $\mathcal{U}$ to wild categories of $\mathcal{U}$-valued diagrams (Lemma 22).

3 Background on type theory and colimits

Before describing the main connection and its applications, we need to review the type system we work in. For us, the most important data type of this system is the *ordinary colimit*, i.e., the colimit of a diagram of types over a graph. We define this type in Section 3.3 and call it "ordinary" to distinguish it from the notion of *coslice colimit*. The latter takes place in coslices of a universe rather than the universe itself, and we will construct coslice colimits out of ordinary colimits.

3.1 Type system

We assume the reader is familiar with MLTT and HITs in the style of [27]. We will work in MLTT augmented with ordinary colimits and denote this system by $\mathsf{MLTT} + \mathsf{Colim}$. In fact, we need only augment MLTT with pushouts as they let us construct all nonrecursive 1-HITs, including ordinary colimits, with all of their computational properties. Notably, $\mathsf{MLTT} + \mathsf{Colim}$ comes with strong function extensionality for free. This property is critical for reasoning about functions in type theory and underlies our entire development (see, for example, Lemma 8). Overall, we carry out our proofs inside $\mathsf{MLTT} + \mathsf{Colim}$ until Section 7. For Section 7 and Section 8, we also assume Voevodsky's univalence axiom.

Before reviewing ordinary colimits, we recall two essential constructions in our type system. The first is the function $\mathsf{ap} : (x = y) \to (f(x) = f(y))$ defined by path induction for all functions $f : X \to Y$ and $x, y : X$. (We use $=$ for the identity type and $\equiv$ for definitional equality.) If we view X as an ∞-groupoid, then ap is the action of f on morphisms of X (thereby exhibiting f as a functor). The second is the *transport* function $\mathsf{transp}^Y : \prod_{x,y:X} \prod_{p:x=y} Y(x) \to Y(y)$ for any type family Y over X. This notion also gives us a dependent version of ap: If $f : \prod_{x:X} Y(x)$, then we have a function $\mathsf{apd}_f : \prod_{x,y:X} \prod_{p:x=y} \mathsf{transp}^Y(p, f(x)) = f(y)$. The transport function is essential for stating the induction principle for HITs, e.g., the colimits in Section 3.3. It also satisfies the following coherence law, which we need for our construction of A-colimits.

▶ **Lemma 1.** *Let $f, g : X \to Y$. For all $x, y : X$, $p : x = y$, and $H : f \sim g$, we have a commuting square of identities*

$$\begin{array}{ccc} f(x) & \overset{H(x)}{\Longrightarrow} & g(x) \\ {\scriptstyle \mathsf{ap}_f(p)}\Big\Downarrow & & \Big\Downarrow{\scriptstyle \mathsf{ap}_g(p)} \\ f(y) & \underset{\mathsf{transp}^{z \mapsto f(z)=g(z)}(p, H(x))}{\Longrightarrow} & g(y) \end{array}$$

Finally, a remark on notation: we may use the Agda notation $(x : X) \to Y(x)$ for the type $\prod_{x:X} Y(x)$ for any type family Y over X.

3.2 Graphs

Let $\mathcal{U}$ be a universe and $A : \mathcal{U}$. In classical 1-category theory, a diagram in a category $\mathcal{C}$ is a functor $F : \mathcal{I} \to \mathcal{C}$, where $\mathcal{I}$ is the shape of the diagram. As long as $\mathcal{C}$ is cocomplete, we can form the functor $\mathsf{colim}_{\mathcal{I}}$ sending each diagram over $\mathcal{I}$ to its colimit in $\mathcal{C}$. We, however, want the colimit of a diagram in the ∞-category $A/\mathcal{U}$. This requires the diagram to be an ∞-functor: the functor laws must satisfy coherence laws up to homotopy, which themselves must satisfy higher coherence laws, and so on at arbitrarily high levels. It is unknown whether such ∞-functors are definable in HoTT.

To avoid infinite coherence conditions, we specialize $\mathcal{I}$ to the free category generated by a *graph*. A graph Γ is a pair (Γ_0, Γ_1) consisting of a type $\Gamma_0 : \mathcal{U}$ of vertices and a family $\Gamma_1 : \Gamma_0 \to \Gamma_0 \to \mathcal{U}$ of edges. A Γ*-shaped diagram F in $A/\mathcal{U}$* is a pair (F_0, F_1) consisting of a function $F_0 : \Gamma_0 \to A/\mathcal{U}$ and a family of maps $F_1 : (i, j : \Gamma_0) \to \Gamma_1(i, j) \to F_0(i) \to_A F_0(j)$. We may write F for F_0 and F_1. The induced diagram in $A/\mathcal{U}$ satisfies all the infinite coherence laws because its domain is freely generated by the points and edges of Γ.

▶ **Example 2.** For each graph Γ and $D : \mathsf{Ob}(A/\mathcal{U})$, the *constant diagram* $\mathsf{const}_\Gamma(D)$ *at* D is defined by $(\mathsf{const}_\Gamma(D))_0\,(i) :\equiv D$ and $(\mathsf{const}_\Gamma(D))_1\,(i, j, g) :\equiv \mathsf{id}_D$. We often refer to $\mathsf{const}_\Gamma(D)$ simply by D.

We will see that A-colimits interact nicely with *trees*. A tree is a graph without non-directed cycles. Formally, a graph Γ is a *tree* if the quotient Γ_0/Γ_1 is contractible. Both $\mathbb{N}$ and $\mathbb{Z}$ are trees when equipped with the successor ordering:

$$\mathbb{N} \;\equiv\; 0 \longrightarrow 1 \longrightarrow 2 \longrightarrow \cdots \qquad \mathbb{Z} \;\equiv\; \cdots \longrightarrow -1 \longrightarrow 0 \longrightarrow 1 \longrightarrow \cdots$$

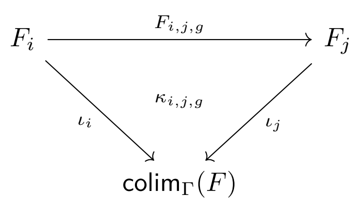
Another example of a tree is a span $\bullet \leftarrow \bullet \rightarrow \bullet$, on which pushouts are defined.

Rijke has defined the notion of *directed tree* and has defined an interpretation function sending an element of a W-type to a directed tree [23, The underlying trees of elements of W-types]. Intuitively, a directed tree is a rooted graph such that for each vertex v, there is a single directed path (including the trivial path) from v to the root. Every directed tree is a tree in our sense [6, Corollary 4.0.6]. Thus, elements of a W-type can be realized as trees.

3.3 Colimits in $\mathcal{U}$

Let F be a Γ-shaped diagram in $\mathcal{U}$. The *(ordinary) colimit of F* is the HIT $\mathsf{colim}_\Gamma(F)$ generated by

$$\begin{aligned} \iota &: (i : \Gamma_0) \to F_i \to \mathsf{colim}_\Gamma(F) \\ \kappa &: (i, j : \Gamma_0)\,(g : \Gamma_1(i, j)) \to \iota_j \circ F_{i,j,g} \sim \iota_i \end{aligned}$$



These two constructors make $\mathsf{colim}_\Gamma(F)$ a *cocone under F* (or *F-cocone*): a type C equipped with maps $u : \prod_{i:\Gamma_0} F_i \to C$ and homotopies $K : \prod_{i,j,g} u_j \circ F_{i,j,g} \sim u_i$. What characterizes $\mathsf{colim}_\Gamma(F)$ as a colimit of F is that κ is a *(homotopy) initial* F-cocone [24]. Equivalently, for every $X : \mathcal{U}$, the function

$$\begin{aligned} \mathsf{postcomp} &: (\mathsf{colim}_\Gamma(F) \to X) \to \mathsf{Cocone}_F(X) \\ \mathsf{postcomp}(f) &:\equiv \big(\lambda i. f \circ \iota_i, \lambda i \lambda j \lambda g \lambda (x : F_i).\, \mathsf{ap}_f(\kappa_{i,j,g}(x))\big) \end{aligned}$$

is an equivalence, where $\mathsf{Cocone}_F(X)$ denotes the type of F-cocones on X.

Our proof of Theorem 15 will use the induction principle for $\mathsf{colim}_\Gamma(F)$. This states that for every type family E over $\mathsf{colim}_\Gamma(F)$ together with data

$$r \;:\; \prod_{i:\Gamma_0}\prod_{x:F_i} E(\iota_i(x)) \qquad R \;:\; \prod_{i,j:\Gamma_0}\;\prod_{g:\Gamma_1(i,j)}\;\prod_{x:F_i} \mathsf{transp}^E(\kappa_{i,j,g}(x), r(j, F_{i,j,g}(x))) = r(i,x)$$

we have a function $\mathsf{ind}(E,r,R) : \prod_{z:\mathsf{colim}_\Gamma(F)} E(z)$ that satisfies $\mathsf{ind}(E,r,R)(\iota_i(x)) \equiv r(i,x)$ and is equipped with an identity $\rho_{\mathsf{ind}(E,r,R)}(i,j,g,x) : \mathsf{apd}_{\mathsf{ind}(E,r,R)}(\kappa_{i,j,g}(x)) = R(i,j,g,x)$.

4 Wild categories

Any universe, along with its coslices, fits into the framework of *wild categories*. This is one approach to category theory in HoTT and is used by other works of synthetic homotopy theory [3, 5, 11]. It is useful for the relationship between A-colimits and factorization systems we establish in Section 7. This requires us to formulate factorization systems on categories other than universes, namely the category of type-valued diagrams over a graph.

The key distinction between wild categories and (pre-)categories [27, Chapter 9.1] is that the latter have 0-truncated hom types. This means that instead of trivializing the higher coherence data for morphisms, wild categories simply ignore them. We choose them over pre-categories because we will focus on universes and their coslices (see Example 7), which are wild categories but not pre-categories.

▶ **Definition 3** ([6, Definition 3.1.1])**.** *A* wild category (in a universe $\mathcal{U}$) *is a tuple consisting of a type* $\mathsf{Ob} : \mathcal{U}$ *of objects, a family* $\mathsf{hom} : \mathsf{Ob} \to \mathsf{Ob} \to \mathcal{U}$ *of hom types, identity morphisms* id*, composition* $\circ$*, left* LId *and right* RId *unit laws for* $\circ$*, and associativity laws* assoc *for* $\circ$*.*

By itself, the data of a wild category is insufficient for our work on factorization systems. We need two extra ingredients. The first is the notion of a wild bicategory. The second is a wild-categorical version of univalence.

▶ **Definition 4.** *A wild category* $\mathcal{C}$ *is a* (wild) bicategory *if it is equipped with identities*

(a) $\mathsf{ap}_{-\circ f}(\mathsf{assoc}(k,g,h)) \cdot \mathsf{assoc}(k, g\circ h, f) \cdot \mathsf{ap}_{k\circ -}(\mathsf{assoc}(g,h,k)) =$ $\mathsf{assoc}(k\circ g, h, f) \cdot \mathsf{assoc}(k, g, h\circ f)$ *for all composable morphisms* k*,* g*,* h*, and* f

(b) $\mathsf{assoc}(g, \mathsf{id}, h) \cdot \mathsf{ap}_{g\circ -}(\mathsf{LId}(h)) = \mathsf{ap}_{-\circ h}(\mathsf{RId}(g))$ *for all composable morphisms* g *and* h.[3]

▶ Remark. For us, a bicategory is always a wild $(2,1)$-category since the 2-cells, which are identities in $\mathcal{U}$, are invertible.

Before moving to univalence, we transfer a well-known lemma of classical 2-category theory to type theory. This was first proved for monoidal categories [10], but the proof is applicable to all bicategories. (The type-theoretic version also appears as [3, Lemma 4.3].)

▶ **Lemma 5** ([6, Lemma 3.1.3])**.** *Let* $\mathcal{C}$ *be a bicategory. For all* $A, B, C : \mathsf{Ob}(\mathcal{C})$*,* $f : \mathsf{hom}_\mathcal{C}(A,B)$*,* $g : \mathsf{hom}_\mathcal{C}(B,C)$*, we have* $\mathsf{LId}(g\circ f)^{-1} \cdot \mathsf{assoc}(\mathsf{id}, g, f)^{-1} \cdot \mathsf{ap}_{-\circ f}(\mathsf{LId}(g)) = \mathsf{refl}_{g\circ f}$.

▶ **Definition 6.** *We say that a wild category* $\mathcal{C}$ *is* univalent *if the canonical function* $(A =_{\mathsf{Ob}(\mathcal{C})} B) \to (A \simeq_\mathcal{C} B)$ *is an equivalence. Here, elements of the righthand type are equivalences, defined as bi-invertible morphisms (in the manner of [27, Definition 4.3.1]).*

[3] A wild bicategory is called a *wild 2-precategory* by [3].

▶ **Example 7.** The following are univalent bicategories assuming the univalence axiom.

- The category $\mathcal{U}$ of types and functions
- For each $A : \mathcal{U}$, the coslice $A/\mathcal{U}$ of $\mathcal{U}$ under A
- The category $\mathsf{Diag}(\Gamma, A/\mathcal{U})$ of Γ-shaped diagrams in $A/\mathcal{U}$. We define its hom types (natural transformations) when we present the action of the A-colimit on maps (Section 5.4).

Our ultimate interest is in colimits in the wild category $A/\mathcal{U}$. This category is defined by

$$\mathsf{Ob}(A/\mathcal{U}) \;:\equiv\; \textstyle\sum_{X:\mathcal{U}} A \to X \qquad\qquad \mathsf{hom}_{A/\mathcal{U}}(X, Y) \;:\equiv\; X \to_A Y$$

For each $X : \mathsf{Ob}(A/\mathcal{U})$, the identity morphism on X is $\big(\mathsf{id}_{\mathsf{pr}_1(X)}, \lambda a.\mathsf{refl}_{\mathsf{pr}_2(X)(a)}\big)$. Composition is defined by $(g, g_p) \circ (f, f_p) :\equiv \big(g \circ f, \lambda a.\, \mathsf{ap}_g(f_p(a)) \cdot g_p(a)\big)$. The associativity and unit laws follow from routine path algebra. Note that the categories $\mathbf{0}/\mathcal{U}$ and $\mathcal{U}$ are equivalent.

We write ty and str for the functions $\mathsf{pr}_1 : \mathsf{Ob}(A/\mathcal{U}) \to \mathcal{U}$ and $\mathsf{pr}_2 : (Z : \mathsf{Ob}(A/\mathcal{U})) \to A \to \mathsf{pr}_1(Z)$, i.e., the underlying type and structure map of an object in $A/\mathcal{U}$, respectively. Also, we write fun and pt for the functions $\mathsf{pr}_1 : \mathsf{hom}_{A/\mathcal{U}}(W, Z) \to \mathsf{ty}(W) \to \mathsf{ty}(Z)$ and $\mathsf{pr}_2 : \big(h : \mathsf{hom}_{A/\mathcal{U}}(W, Z)\big) \to \mathsf{pr}_1(h) \circ \mathsf{str}(W) \sim \mathsf{str}(Z)$, respectively.

▶ **Lemma 8.** *Let $f, g : X \to_A Y$. Define $f \sim_A g$ as the type of homotopies $H : \mathsf{fun}(f) \sim \mathsf{fun}(g)$ equipped with a commuting triangle*

$$\begin{array}{ccc}
\mathsf{fun}(f)(\mathsf{str}(X)(a)) & \xRightarrow{\;\mathsf{pt}(f)(a)\;} & \mathsf{str}(Y)(a) \\
{\scriptstyle H(\mathsf{str}(X)(a))}\Big\Downarrow & \nearrow\!\!\!\nearrow {\scriptstyle \mathsf{pt}(g)(a)} & \\
\mathsf{fun}(g)(\mathsf{str}(X)(a)) & &
\end{array}$$

for each $a : A$. The canonical function $f = g \to f \sim_A g$ is an equivalence, with inverse denoted by $\langle -, - \rangle : f \sim_A g \to f = g$.

Elements of $f \sim_A g$ are called *A-homotopies* between f and g.

4.1 Orthogonal factorization systems

We now introduce (orthogonal) factorization systems on wild categories. For us, the key property of such systems is that they interact nicely with adjunctions. In Section 7, we deduce from this property, combined with the main connection, that A-colimits preserve the left classes of factorization systems on $\mathcal{U}$.

▶ **Definition 9.** *Let $\mathcal{C}$ be a wild category. An* orthogonal factorization system *(OFS) on $\mathcal{C}$ consists of predicates $\mathcal{L}, \mathcal{R} : \prod_{A,B:\mathcal{C}} \mathsf{hom}_{\mathcal{C}}(A, B) \to \mathsf{Prop}$ such that*

1. *both $\mathcal{L}$ and $\mathcal{R}$ are closed under composition and have all identities*
2. *for every $h : \mathsf{hom}_{\mathcal{C}}(A, B)$, the following type is contractible:*

$$\mathsf{fact}_{\mathcal{L},\mathcal{R}}(h) \;:\equiv\; \sum_{D:\mathsf{Ob}(\mathcal{C})} \; \sum_{f:\mathsf{hom}_{\mathcal{C}}(A,D)} \; \sum_{g:\mathsf{hom}_{\mathcal{C}}(D,B)} (g \circ f = h) \times \mathcal{L}(f) \times \mathcal{R}(g)$$

For the next lemma, where $\mathcal{C}$ is a univalent bicategory, $\mathcal{C}$ is similar enough to $\mathcal{U}$ that the proof of the lemma for $\mathcal{U}$ can be transferred to $\mathcal{C}$.[4] Indeed, univalence lets us characterize the identity types of $\mathsf{fact}_{\mathcal{L},\mathcal{R}}(h)$ via the fundamental theorem of identity types [21, Theorem 11.2.2]. Moreover, Lemma 5 gives us a suitable diagonal filler for the key commuting square used by the proof. Before stating the next lemma, we need a definition.

[4] For the proof of this lemma for $\mathcal{U}$, see [22, Lemma 1.46].

▶ **Definition 10.** *Let $\mathcal{C}$ be a wild category. Let $l : \mathsf{hom}_{\mathcal{C}}(A, B)$ and $\mathcal{H}$ be a property of morphisms in $\mathcal{C}$. We say that l has the* left lifting property *against $\mathcal{H}$ if for every $r : \mathsf{hom}_{\mathcal{C}}(C, D)$ such that $\mathcal{H}(r)$ and every commuting square*

$$\begin{array}{ccc} A & \xrightarrow{f} & C \\ {\scriptstyle l}\downarrow & S & \downarrow{\scriptstyle r} \\ B & \xrightarrow[g]{} & D \end{array}$$

the type of diagonal fillers

$$\sum_{d:\mathsf{hom}_{\mathcal{C}}(B,C)} \sum_{H_f : d \circ l = f} \sum_{H_g : r \circ d = g} \mathsf{assoc}(r, d, l) \cdot \mathsf{ap}_{r \circ -}(H_f) = \mathsf{ap}_{- \circ l}(H_g) \cdot S$$

is contractible.

▶ **Lemma 11** ([6, Corollary 3.3.6])**.** *Suppose that $\mathcal{C}$ is a univalent bicategory with an OFS $(\mathcal{L}, \mathcal{R})$. A map is in $\mathcal{L}$ if and only if it has the left lifting property against $\mathcal{R}$.*

This alternative definition of $\mathcal{L}$ is useful for the proof of Theorem 13, below. For this theorem, we need to introduce adjoint pairs of functors between wild categories.

▶ **Definition 12.** *Let $L : \mathcal{C} \to \mathcal{D}$ and $R : \mathcal{D} \to \mathcal{C}$ be functors of wild categories. An* adjunction *$L \dashv R$ consists of an equivalence $\alpha : \mathsf{hom}_{\mathcal{D}}(LA, X) \simeq \mathsf{hom}_{\mathcal{C}}(A, RX)$ for all $A : \mathsf{Ob}(\mathcal{C})$ and $X : \mathsf{Ob}(\mathcal{D})$ along with naturality proofs:*

$$\mathsf{n}_1 \;:\; \prod_{A:\mathsf{Ob}(\mathcal{C})} \prod_{X,Y:\mathsf{Ob}(\mathcal{D})} \prod_{g:\mathsf{hom}_{\mathcal{D}}(X,Y)} \prod_{h:\mathsf{hom}_{\mathcal{D}}(LA,X)} R(g) \circ \alpha(h) = \alpha(g \circ h)$$

$$\mathsf{n}_2 \;:\; \prod_{Y:\mathsf{Ob}(\mathcal{D})} \prod_{A,B:\mathsf{Ob}(\mathcal{C})} \prod_{f:\mathsf{hom}_{\mathcal{C}}(A,B)} \prod_{h:\mathsf{hom}_{\mathcal{D}}(LB,Y)} \alpha(h) \circ f = \alpha(h \circ L(f))$$

▶ **Theorem 13** ([6, Corollary 3.3.9])**.** *Consider an adjunction $L \dashv R$ where both $\mathcal{C}$ and $\mathcal{D}$ are univalent bicategories. If R preserves $\mathcal{R}$, then L preserves $\mathcal{L}$.*

5 The main connection

Let $\mathcal{U}$ be a universe. Let Γ be a graph and suppose F is a diagram in $A/\mathcal{U}$ over Γ. Working in $\mathsf{MLTT} + \mathsf{Colim}$, we want to construct the A-*colimit* of F so as to show the connection between A-colimits and ordinary colimits. After defining A-*colimit*, we mention a reasonable yet wrong approach to constructing it. Then, we explain another construction and prove it is correct by exhibiting it as left adjoint to the constant diagram functor. The Agda proof of this adjunction is found in the folder [7, `Colimit-code/Main-Theorem`].

5.1 Definition of A-colimits

We can generalize ordinary colimits in Section 3 to all coslices $A/\mathcal{U}$. For each $Y : \mathsf{Ob}(A/\mathcal{U})$, an F-cocone on Y consists of a family of maps $h : (i : \Gamma_0) \to F_i \to_A Y$ in $A/\mathcal{U}$ together with an identity $H_{i,j,g} \;:\; h_j \circ F_{i,j,g} = h_i$ for all $i, j : \Gamma_0$ and $g : \Gamma_1(i, j)$. In this situation, we say that Y is a *colimit of* F if (h, H) is initial in the category of F-cocones. This means that for each $X : \mathsf{Ob}(A/\mathcal{U})$, the function

$$\begin{aligned} \mathsf{postcomp}(h, H) \;&:\; (Y \to_A X) \;\to\; \mathsf{Cocone}_F(X) \\ \mathsf{postcomp}(h, H, f) \;&:\equiv\; \big(\lambda i. f \circ h_i, \lambda i \lambda j \lambda g. \mathsf{assoc}(f, h_j, F_{i,j,g}) \cdot \mathsf{ap}_{f \circ -}(H_{i,j,g})\big) \end{aligned}$$

is an equivalence. We must include the associativity term since associativity of maps does not hold judgmentally in $A/\mathcal{U}$ (whereas it does in $\mathcal{U}$).

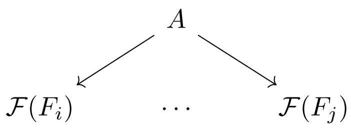
Observe that by a variant of Lemma 8, $h_j \circ F_{i,j,g} = h_i$ is equivalent to the type of homotopies $\eta_{i,j,g} : \mathsf{fun}(h_j) \circ \mathsf{fun}(F_{i,j,g}) \sim \mathsf{fun}(h_i)$ equipped with a commuting square

$$\begin{array}{ccc} \mathsf{fun}(h_j)(\mathsf{fun}(F_{i,j,g})(\mathsf{str}(F_i)(a))) & \xRightarrow{\eta_{i,j,g}(\mathsf{str}(F_i)(a))} & \mathsf{fun}(h_i)(\mathsf{str}(F_i)(a)) \\ {\scriptstyle \mathsf{ap}_{\mathsf{fun}(h_j)}(\mathsf{pt}(F_{i,j,g})(a))} \Big\Downarrow & & \Big\Downarrow {\scriptstyle \mathsf{pt}(h_i)(a)} \\ \mathsf{fun}(h_j)(\mathsf{str}(F_j)(a)) & \xRightarrow[\mathsf{pt}(h_j)(a)]{} & \mathsf{str}(Y)(a) \end{array} \tag{2-c}$$

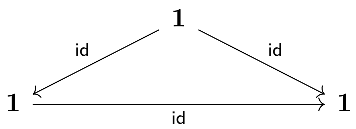
of paths for each $a : A$. It is this family of 2-cells which distinguishes the colimit of F, in $A/\mathcal{U}$, from $\mathsf{colim}_\Gamma(\mathcal{F}(F))$. Here, we reuse $\mathcal{F}$ to denote the evident forgetful functor from Γ-shaped diagrams in $A/\mathcal{U}$ to those in $\mathcal{U}$. The 2-cells affect $\mathsf{colim}_\Gamma(\mathcal{F}(F))$ by collapsing its nontrivial loops formed by paths of the form $\eta(\mathsf{str}(F_i)(a))$. We call such loops *distinguished loops* in $\mathsf{colim}_\Gamma(\mathcal{F}(F))$. For example, if $i \equiv j$ and $F_{i,j,g} \equiv \mathsf{id}_{F_i}$, then (2-c) is equivalent to $\eta(\mathsf{str}(F_i)(a)) = \mathsf{refl}_{\mathsf{fun}(h_i)(\mathsf{str}(F_i)(a))}$. In this case, it fills the loop $\eta(\mathsf{str}(F_i)(a))$.

5.2 Misleading approach

If our setting behaved like the classical one, the colimit of F in $A/\mathcal{U}$ would arise as the ordinary colimit of the *augmented diagram*: $\mathcal{F}(F)$ augmented with the canonical arrow from A to $\mathcal{F}(F_i)$ for each $i : \Gamma_0$ [17, Proposition 4.6]. If Γ is *discrete*, i.e., Γ_1 is the empty relation, then the A-colimit of F inside HoTT is, in fact, the colimit of



In general, though, this construction is wrong inside HoTT. For example, the pointed colimit of the diagram $\mathbf{1} \xrightarrow{\mathsf{id}} \mathbf{1}$ is trivial, but the colimit of the augmented diagram

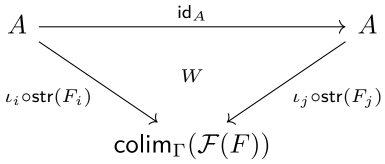


is the circle S^1. The reason for the discrepancy between the classical case and ours is that unless Γ is discrete, the augmented diagram inside HoTT adds arrows that are intended as composites but are not interpreted as such in the model of HoTT. Rather, the model sees them as freely added to the diagram.

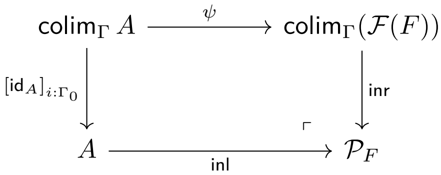
5.3 Our approach

Our approach to building the colimit of F never creates an augmented diagram, thereby avoiding the problem of Section 5.2. We start with the ordinary colimit $\mathsf{colim}_\Gamma(\mathcal{F}(F))$ which ignores the coslice structure of F. Then, we glue onto this colimit the 2-cells required by the coslice colimit. We do this via a quotient of $\mathsf{colim}_\Gamma(\mathcal{F}(F))$ that fills its distinguished loops.

To this end, define $\mathsf{colim}_\Gamma\, A \xrightarrow{\psi} \mathsf{colim}_\Gamma(\mathcal{F}(F))$ by colimit induction, as the function induced by the cocone



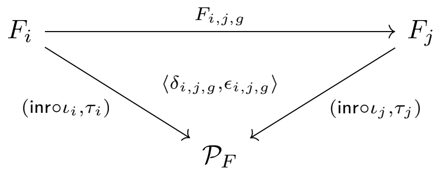
under the constant diagram at A. The homotopy $W : \iota_j \circ \mathsf{str}(F_j) \sim \iota_i \circ \mathsf{str}(F_i)$ is defined by $W(a) :\equiv \mathsf{ap}_{\iota_j}(\mathsf{pt}(F_{i,j,g})(a))^{-1} \cdot \kappa_{i,j,g}(\mathsf{str}(F_i)(a))$. Intutively, ψ finds the distinguished loops of $\mathsf{colim}_\Gamma(\mathcal{F}(F))$. Next, form the pushout square



which, by the definition of pushout types, comes with a homotopy $\mathsf{glue}_{\mathcal{P}_F} : \mathsf{inl} \circ [\mathsf{id}_A] \sim \mathsf{inr} \circ \psi$. This pushout is our approach to forming the desired quotient of $\mathsf{colim}_\Gamma(\mathcal{F}(F))$.

▶ **Example 14.** Suppose that Γ has a single vertex v and a single loop ℓ at v. Let $\mathbf{2}$ denote the type of booleans. Define the pointed diagram F over Γ by $F_v :\equiv (\mathbf{2}, \mathsf{true})$ and $F_{v,v,\ell} :\equiv (\mathsf{id}_{\mathbf{2}}, \mathsf{refl})$. Then $\mathsf{colim}_\Gamma(\mathcal{F}(F)) \simeq S^1 + S^1$. In this case, the function ψ traces the left copy of S^1, the distinguished loop, exactly once. The pushout $\mathcal{P}_F$ is formed from $S^1 + S^1$ by filling this loop, which collapses the left copy of S^1 to a point. As a result, $\mathcal{P}_F \simeq \mathbf{1} + S^1$.

Back to the general case, with the equivalence $\langle -, - \rangle$ of Lemma 8, we can form an F-cocone on $(\mathcal{P}_F, \mathsf{inl})$



$$(\tau_i(a) :\equiv \mathsf{glue}_{\mathcal{P}_F}(\iota_i(a))^{-1})$$

as follows. We have a homotopy $\delta_{i,j,g} :\equiv \lambda(x : \mathsf{ty}(F_i)).\, \mathsf{ap}_{\mathsf{inr}}(\kappa_{i,j,g}(x))$ from $\mathsf{inr} \circ \iota_j \circ \mathsf{fun}(F_{i,j,g})$ to $\mathsf{inr} \circ \iota_i$. Further, for each $a : A$, we have a chain $\epsilon_{i,j,g}(a)$ of identities

$$\begin{aligned}
& \mathsf{ap}_{\mathsf{inr}}(\kappa_{i,j,g}(\mathsf{str}(F_i)(a)))^{-1} \cdot \mathsf{ap}_{\mathsf{inr}\circ\iota_j}(\mathsf{pt}(F_{i,j,g})(a)) \cdot \tau_j(a) \\
= \;& \mathsf{ap}_{\mathsf{inr}}(\mathsf{ap}_{\iota_j}(\mathsf{pt}(F_{i,j,g})(a))^{-1} \cdot \kappa_{i,j,g}(\mathsf{str}(F_i)(a)))^{-1} \cdot \tau_j(a) \cdot \mathsf{refl}_{\mathsf{inl}(a)} \\
= \;& \mathsf{ap}_{\mathsf{inr}}(\mathsf{ap}_{\psi}(\kappa_{i,j,g}(a)))^{-1} \cdot \tau_j(a) \cdot \mathsf{refl}_{\mathsf{inl}(a)} && (\textit{via } \rho_\psi(i,j,g,a)) \\
= \;& \mathsf{ap}_{\mathsf{inr}}(\mathsf{ap}_{\psi}(\kappa_{i,j,g}(a)))^{-1} \cdot \tau_j(a) \cdot \mathsf{ap}_{\mathsf{inl}}(\mathsf{ap}_{[\mathsf{id}_A]}(\kappa_{i,j,g}(a))) && (\textit{via } \rho_{[\mathsf{id}_A]}(i,j,g,a)) \\
= \;& \mathsf{transp}^{\mathsf{inr}\circ\psi\sim\mathsf{inl}\circ[\mathsf{id}_A]}(\kappa_{i,j,g}(a), \tau_j(a)) && (\textit{Lemma } 1) \\
= \;& \tau_i(a) && (\textit{by } \mathsf{apd}_{\mathsf{glue}(-)^{-1}}(\kappa_{i,j,g}(a)))
\end{aligned}$$

Let $\mathcal{K}(\mathcal{P}_F)$ denote this F-cocone structure on $(\mathcal{P}_F, \mathsf{inl})$.

▶ **Theorem 15** ([6, Theorem 5.4.3])**.** *The function*

$$\mathsf{postcomp}(\mathcal{K}(\mathcal{P}_F), T, f_T) \;:\; ((\mathcal{P}_F, \mathsf{inl}) \to_A (T, f_T)) \;\to\; \mathsf{Cocone}_F(T, f_T)$$

is an equivalence for every $(T, f_T) : \mathsf{Ob}(A/\mathcal{U})$.

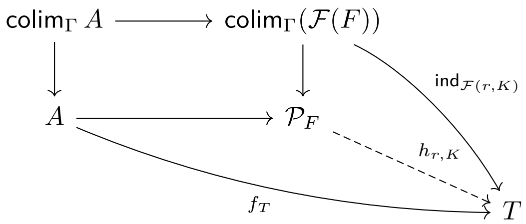
Proof. We construct an inverse $\mathsf{Cocone}_F(T, f_T) \xrightarrow{\Theta} ((\mathcal{P}_F, \mathsf{inl}) \to_A (T, f_T))$ of $\mathsf{postcomp}(\mathcal{K}(\mathcal{P}_F), T, f_T)$ as follows. Let $(r, K) : \mathsf{Cocone}_F(T, f_T)$. The forgetful functor $\mathcal{F}$ from cocones under F to ordinary cocones under $\mathcal{F}(F)$ gives rise to the function $\mathsf{ind}_{\mathcal{F}(r,K)} : \mathsf{colim}_\Gamma(\mathcal{F}(F)) \to T$ by colimit induction. For all $i : \Gamma_0$ and $a : A$, we have

$$f_T(a) \overset{\mathsf{pt}(r_i)(a)^{-1}}{=\!=\!=\!=} \mathsf{fun}(r_i)(\mathsf{str}(F_i(a))) =\!=\!= \mathsf{ind}_{\mathcal{F}(r,K)}(\mathsf{str}(F_i(a)))$$

Further, for all $i, j : \Gamma_0$, $g : \Gamma_1(i, j)$, and $a : A$, we have a chain of identities

$$\begin{aligned}
&\mathsf{transp}^{f_T \circ [\mathsf{id}_A] = \mathsf{ind}_{\mathcal{F}(r,K)} \circ \psi(x)}(\kappa_{i,j,g}(a), \mathsf{pr}_2(r_j)(a)^{-1}) \\
&= \mathsf{ap}_{f_T}(\mathsf{ap}_{[\mathsf{id}_A]}(\kappa_{i,j,g}(a)))^{-1} \cdot \mathsf{pr}_2(r_j)(a)^{-1} \cdot \mathsf{ap}_{\mathsf{ind}_{\mathcal{F}(r,K)}}(\mathsf{ap}_\psi(\kappa_{i,j,g}(a))) && (\textit{Lemma } 1) \\
&= \mathsf{ap}_{f_T}(\mathsf{ap}_{[\mathsf{id}_A]}(\kappa_{i,j,g}(a)))^{-1} \cdot \mathsf{pr}_2(r_j)(a)^{-1} \cdot \mathsf{ap}_{\mathsf{pr}_1(r_j)}(\mathsf{pt}(F_{i,j,g})(a))^{-1} \cdot \mathsf{pr}_1(K_{i,j,g})(\mathsf{str}(F_i)(a)) \\
& && (\textit{via } \rho_\psi(i, j, g, a) \textit{ and then } \rho_{\mathsf{ind}_{\mathcal{F}(r,K)}}(i, j, g, \mathsf{str}(F_i)(a))) \\
&= \left(\mathsf{pr}_1(K_{i,j,g})(\mathsf{str}(F_i)(a))^{-1} \cdot \mathsf{ap}_{\mathsf{pr}_1(r_j)}(\mathsf{pt}(F_{i,j,g})(a)) \cdot \mathsf{pr}_2(r_j)(a)\right)^{-1} && (\textit{via } \rho_{[\mathsf{id}_A]}(i, j, g, a)) \\
&= \mathsf{pr}_2(r_i)(a)^{-1} && (\textit{by } \mathsf{ap}_{_^{-1}}(\mathsf{pr}_2(K_{i,j,g})(a)))
\end{aligned}$$

By induction on $\mathsf{colim}_\Gamma A$, this gives us a homotopy $f_T \circ [\mathsf{id}_A] \sim \mathsf{ind}_{\mathcal{F}(r,K)} \circ \psi$ and thus a function $h_{r,K} : \mathcal{P}_F \to T$



defined by pushout induction on $\mathcal{P}_F$. Finally, since $h(\mathsf{inl}(a)) \equiv f_T(a)$ for all $a : A$, we have

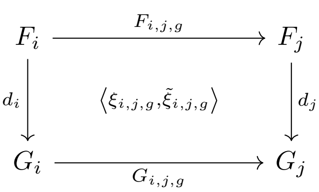
$$\Theta(r, K) :\equiv \left(h_{r,K}, \lambda a.\mathsf{refl}_{f_T(a)}\right) \ : \ (\mathcal{P}_F, \mathsf{inl}) \to_A (T.f_T)$$

Each of the homotopies $\mathsf{postcomp}(\mathcal{K}(\mathcal{P}_F), T, f_T) \circ \Theta \sim \mathsf{id}$ and $\Theta \circ \mathsf{postcomp}(\mathcal{K}(\mathcal{P}_F), T, f_T) \sim \mathsf{id}$ requires intricate computations to prove. We leave their proofs to the formalization (see the folders [7, `Colimit-code/R-L-R`] and [7, `Colimit-code/L-R-L`], respectively). ◀

5.4 Action on maps

So far, we have defined a function $\mathsf{colim}^A_\Gamma :\equiv \mathcal{P} : \mathsf{Ob}(\mathsf{Diag}(\Gamma, A/\mathcal{U})) \to \mathsf{Ob}(A/\mathcal{U})$ sending a Γ-shaped diagram in $A/\mathcal{U}$ to its A-colimit. We now make $\mathcal{P}$ a functor by describing its action on maps of diagrams. We want to describe this action in terms of the action of the ordinary colimit functor by using the special form of $\mathcal{P}$'s object function. Moreover, we must verify that such a description is correct by proving that $\mathcal{P}$ is left adjoint to the constant diagram functor, i.e., enjoys the universal property of the colimit functor.

Suppose that F and G are Γ-shaped diagrams in $A/\mathcal{U}$. The type of *natural transformations* from F to G consists of families $d : (i : \Gamma_0) \to \mathsf{ty}(F_i) \to_A \mathsf{ty}(G_i)$ of maps equipped with an A-homotopy $G_{i,j,g} \circ d_i \sim_A d_j \circ F_{i,j,g}$ for all i, j, g, where $\sim_A$ is as in Lemma 8. Consider a natural transformation $\delta :\equiv \left(d, \langle \xi, \tilde{\xi} \rangle\right)$



from F to G, where $\langle -,- \rangle$ is as in Lemma 8. We form a map $\mathsf{colim}^A_\Gamma(\delta) : \mathsf{colim}^A_\Gamma(F) \to_A \mathsf{colim}^A_\Gamma(G)$ as follows. Start with the function $\mathsf{colim}_\Gamma(\mathcal{F}(F)) \xrightarrow{\bar{\delta}} \mathsf{colim}_\Gamma(\mathcal{F}(G))$ induced by the following map of $\mathcal{U}$-valued diagrams over Γ:

$$\begin{array}{ccc} \mathsf{ty}(F_i) & \xrightarrow{\mathsf{fun}(F_{i,j,g})} & \mathsf{ty}(F_j) \\ {\scriptstyle \mathsf{fun}(d_i)}\downarrow & \xi_{i,j,g} & \downarrow{\scriptstyle \mathsf{fun}(d_j)} \\ \mathsf{ty}(G_i) & \xrightarrow[\mathsf{fun}(G_{i,j,g})]{} & \mathsf{ty}(G_j) \end{array}$$

Note that for each $a : A$,

$$\tilde{\xi}_{i,j,g}(a) \ : \ \xi_{i,j,g}(\mathsf{str}(F_i)(a))^{-1} \cdot \mathsf{ap}_{\mathsf{fun}(G_{i,j,g})}(\mathsf{pt}(d_i)(a)) \cdot \mathsf{str}(G_{i,j,g})(a) = \mathsf{ap}_{\mathsf{fun}(d_j)}(\mathsf{pt}(F_{i,j,g})(a)) \cdot \mathsf{pt}(d_j)(a)$$

We may assume that $\tilde{\xi}_{i,j,g}(a)$ instead has the equivalent type

$$\xi_{i,j,g}(\mathsf{str}(F_i)(a)) = \underbrace{\mathsf{ap}_{\mathsf{fun}(G_{i,j,g})}(\mathsf{pt}(d_i)(a)) \cdot \mathsf{str}(G_{i,j,g})(a) \cdot \mathsf{pt}(d_j)(a)^{-1} \cdot \mathsf{ap}_{\mathsf{fun}(d_j)}(\mathsf{pt}(F_{i,j,g})(a))^{-1}}_{E_{i,j,g}(a)}$$

Here we abbreviate the right endpoint of the path by $E_{i,j,g}(a)$. Now, the triangle

$$\begin{array}{ccc} & \mathsf{colim}_\Gamma A & \\ {\scriptstyle \psi_F}\swarrow & & \searrow{\scriptstyle \psi_G} \\ \mathsf{colim}_\Gamma(\mathcal{F}(F)) & \xrightarrow[\bar{\delta}]{} & \mathsf{colim}_\Gamma(\mathcal{F}(G)) \end{array} \qquad (\psi\text{-tri})$$

commutes by induction on $\mathsf{colim}_\Gamma A$. Indeed, the computation rules of these functions give us

$$C_i(a) :\equiv \mathsf{ap}_{\iota_i}(\mathsf{pt}(d_i)(a)) \ : \ \bar{\delta}(\psi_F(\iota_i(a))) = \psi_G(\iota_i(a))$$

for all $i : \Gamma_0$ and $a : A$. Further, by defining $\Lambda_{i,j,g}(a) :\equiv C_j(a) \cdot \mathsf{ap}_{\psi_G}(\kappa_{i,j,g}(a))$, we have a chain of identities

$$\begin{aligned}
& \mathsf{transp}^{\bar{\delta}\circ\psi_F \sim \psi_G}(\kappa_{i,j,g}(a), C_j(a)) \\
= \ & \mathsf{ap}_{\bar{\delta}}(\mathsf{ap}_{\psi_F}(\kappa_{i,j,g}(a)))^{-1} \cdot C_j(a) \cdot \mathsf{ap}_{\psi_G}(\kappa_{i,j,g}(a)) && (\textit{Lemma } 1) \\
= \ & \left(\mathsf{ap}_{\iota_j}(\xi_{i,j,g}(\mathsf{str}(F_i)(a)))^{-1} \cdot \kappa_{i,j,g}(\mathsf{fun}(d_i)(\mathsf{str}(F_i)(a)))\right)^{-1} \cdot \mathsf{ap}_{\iota_j \circ \mathsf{fun}(d_j)}(\mathsf{pt}(F_{i,j,g})(a)) \cdot \Lambda_{i,j,g}(a) \\
& && (\textit{via } \rho_{\psi_F}(i,j,g,a) \textit{ and then } \rho_{\bar{\delta}}(i,j,g,\mathsf{str}(F_i)(a))) \\
= \ & \left(\mathsf{ap}_{\iota_j}(E_{i,j,g}(\mathsf{str}(F_i)(a)))^{-1} \cdot \kappa_{i,j,g}(\mathsf{fun}(d_i)(\mathsf{str}(F_i)(a)))\right)^{-1} \cdot \mathsf{ap}_{\iota_j \circ \mathsf{fun}(d_j)}(\mathsf{pt}(F_{i,j,g})(a)) \cdot \Lambda_{i,j,g}(a) \\
& && (\textit{by } \mathsf{ap}_{\left(\mathsf{ap}_{\iota_j}(-)^{-1} \cdot \kappa_{i,j,g}(\mathsf{fun}(d_i)(\mathsf{str}(F_i)(a)))\right)^{-1} \cdot \ldots}(\tilde{\xi}_{i,j,g}(a))) \\
= \ & C_i(a) && (\textit{via } \rho_{\psi_G}(i,j,g,a))
\end{aligned}$$

for all $i, j : \Gamma_0$, $g : \Gamma_1(i,j)$, and $a : A$, so (ψ-tri) commutes. We now have a map

$$\begin{array}{ccccc} A & \longleftarrow & \mathsf{colim}_\Gamma A & \longrightarrow & \mathsf{colim}_\Gamma(\mathcal{F}(F)) \\ {\scriptstyle \mathsf{id}}\downarrow & \lambda x.\mathsf{refl}_{[\mathsf{id}](x)} & \downarrow{\scriptstyle \mathsf{id}} & \lambda x.C(x)^{-1} & \downarrow{\scriptstyle \bar{\delta}} \\ A & \longleftarrow & \mathsf{colim}_\Gamma A & \longrightarrow & \mathsf{colim}_\Gamma(\mathcal{F}(G)) \end{array}$$

of spans, which induces a function $\Psi_\delta : \mathcal{P}_F \to \mathcal{P}_G$ by the universal property of pushouts. Since $\Psi_\delta(\mathsf{inl}(a)) \equiv \mathsf{inl}(a)$ for all $a : A$, we may take $\mathsf{colim}^A_\Gamma(\delta)$ as $\left(\Psi_\delta, \lambda a.\mathsf{refl}_{\mathsf{inl}(a)}\right) : \mathcal{P}_F \to_A \mathcal{P}_G$.

To verify that the functor $\mathsf{Diag}(\Gamma, A/\mathcal{U}) \xrightarrow{\mathsf{colim}^A_\Gamma} A/\mathcal{U}$ we've defined is correct, we must show that it is left adjoint to the constant diagram functor. To do so, we construct the terms n_1 and n_2 required by Definition 12.

▶ **Lemma 16** ([6, Lemma 5.4.5]). *For every map $s : V \to_A U$, the following square commutes:*

$$\begin{array}{ccc} \mathsf{colim}^A_\Gamma(F) \to_A V & \xrightarrow{\;s\circ-\;} & \mathsf{colim}^A_\Gamma(F) \to_A U \\ {\scriptstyle \mathsf{postcomp}(\mathcal{K}(\mathcal{P}_F),V)}\downarrow & & \downarrow{\scriptstyle \mathsf{postcomp}(\mathcal{K}(\mathcal{P}_F),U)} \\ \mathsf{Cocone}_F(V) & \xrightarrow[\mathsf{Cocone}_F(s\circ-)]{} & \mathsf{Cocone}_F(U) \end{array}$$

▶ **Lemma 17** ([6, Lemma 5.4.12]). *For every $V : \mathsf{Ob}(A/\mathcal{U})$ and $\delta : F \Rightarrow_A G$, the following square commutes:*

$$\begin{array}{ccc} \mathsf{colim}^A_\Gamma(G) \to_A V & \xrightarrow{\;-\circ\mathsf{colim}^A_\Gamma(\delta)\;} & \mathsf{colim}^A_\Gamma(F) \to_A V \\ {\scriptstyle \mathsf{postcomp}(\mathcal{K}(\mathcal{P}_G),V)}\downarrow & & \downarrow{\scriptstyle \mathsf{postcomp}(\mathcal{K}(\mathcal{P}_F),V)} \\ \mathsf{Cocone}_G(V) & \xrightarrow[\mathsf{Cocone}^V(-\circ\delta)]{} & \mathsf{Cocone}_F(V) \end{array}$$

The two lower horizontal functions are induced by post-composition with s and pre-composition with δ [6, Definition 5.4.11], respectively.

Lemma 16 is a routine computation, whereas Lemma 17 is quite difficult. The proof of Lemma 17 is easier for the map $\mathsf{colim}^A_\Gamma(F) \to_A \mathsf{colim}^A_\Gamma(G)$ obtained by applying the inverse of $\mathsf{postcomp}(\mathcal{K}(\mathcal{P}_F), \mathcal{P}_G, \mathsf{inl})$ to the canonical F-cocone on $\mathcal{P}_G$ induced by δ. Therefore, we decide to reduce the goal to an equality between this map and $\mathsf{colim}^A_\Gamma(\delta)$. We achieve this by showing that they belong to the same fiber of $\mathsf{postcomp}(\mathcal{K}(\mathcal{P}_F), \mathcal{P}_G, \mathsf{inl})$, which is contractible by Theorem 15. Though much easier than a direct approach to Lemma 17, this method requires intricate computations. We have formalized both Lemma 16 and Lemma 17 (see [7, `Colimit-code/Map-Nat/CosColimitPstCmp.agda`] and [7, `Colimit-code/Map-Nat/CosColimitPreCmp.agda`], respectively).

6 Creation of colimits

Classically, if $\mathcal{D}$ is an ∞-category, then all forgetful functors of ∞-coslices create $\mathcal{D}$-shaped colimits when the ∞-groupoid obtained by freely inverting all morphisms of $\mathcal{D}$ is contractible (see [15, Tag 02KS]). Theorem 18 expresses the same result inside HoTT.

▶ **Theorem 18.** *The forgetful functor $A/\mathcal{U} \to \mathcal{U}$ creates colimits over trees.*

Proof. The idea is that a tree has no cycles, and thus we have no distinguished loops to fill. As a result, coslice colimits over trees look the same as their underlying colimits in $\mathcal{U}$.

To be precise, suppose that Γ is a tree and let F be a diagram in $A/\mathcal{U}$ over Γ. Then the function $[\mathsf{id}_A] : \mathsf{colim}_\Gamma A \to A$ is an equivalence, and one can check that

$$\begin{array}{ccc} \mathsf{colim}_\Gamma A & \xrightarrow{\;\psi\;} & \mathsf{colim}_\Gamma(\mathcal{F}(F)) \\ {\scriptstyle [\mathsf{id}_A]}\downarrow & & \downarrow{\scriptstyle \mathsf{id}} \\ A & \xrightarrow[\psi\circ[\mathsf{id}_A]^{-1}]{} & \mathsf{colim}_\Gamma(\mathcal{F}(F)) \end{array}$$

is a pushout square. By uniqueness of pushouts, this gives us an equivalence $\gamma : \mathcal{P}_F \xrightarrow{\simeq} \mathsf{colim}_\Gamma(\mathcal{F}(F))$ such that $\gamma(\mathsf{inr}(\iota_i(x))) \equiv \iota_i(x)$ for all $i : \Gamma_0$ and $x : \mathsf{ty}(F_i)$. We also see that

$$\mathsf{ap}_\gamma(\mathsf{ap}_{\mathsf{inr}}(\kappa_{i,j,g}(x))) = \mathsf{ap}_{\gamma\circ\mathsf{inr}}(\kappa_{i,j,g}(x)) \equiv \mathsf{ap}_{\mathsf{id}}(\kappa_{i,j,g}(x)) = \kappa_{i,j,g}(x)$$

for all $i, j : \Gamma_0$, $g : \Gamma_1(i, j)$, and $x : \mathsf{ty}(F_i)$. This means that γ is a morphism of cocones under $\mathcal{F}(F)$. It follows that the forgetful functor preserves colimits over Γ.

It remains to prove that the forgetful functor reflects colimits over Γ. Consider an A-cocone $\mathcal{J}$ under F

$$\begin{array}{ccc} F_i & \xrightarrow{F_{i,j,g}} & F_j \\ & \langle H,K\rangle & \\ {\scriptstyle r_i}\searrow & & \swarrow{\scriptstyle r_j} \\ & C & \end{array}$$

as well as the cocone $\mathcal{F}(\mathcal{J}) :\equiv (\mathsf{ty}(C), \mathsf{fun} \circ r, H)$ under $\mathcal{F}(F)$ obtained by applying the forgetful functor to $\mathcal{J}$. Suppose that $\mathcal{F}(\mathcal{J})$ is colimiting in $\mathcal{U}$. By the universal property of colimits in $A/\mathcal{U}$, we have a morphism $(\mathcal{P}_F, \mathsf{inl}) \xrightarrow{\tau} C$ of A-cocones, which induces a morphism $\mathcal{P}_F \xrightarrow{\mathcal{F}(\tau)} \mathsf{ty}(C)$ of cocones in $\mathcal{U}$. This morphism is unique by the universal property of ordinary colimits. Moreover, by the uniqueness of ordinary colimits, there is a cocone equivalence from $\mathcal{P}_F$ to $\mathsf{ty}(C)$ as both of them are colimiting. This implies $\mathcal{F}(\tau)$ is an equivalence. Thus, τ is an A-cocone morphism whose underlying function $\mathcal{P}_F \to \mathsf{ty}(C)$ is an equivalence. This means that τ is an A-cocone equivalence, so that $\mathcal{J}$ is colimiting. ◀

▶ Remark. The fact that the forgetful functor $\mathcal{U}^* \to \mathcal{U}$ from pointed types creates *pushouts* appears in the agda-unimath library, though without proof [23, Pushouts of pointed types].

Theorem 18 lets us lift powerful features of ordinary colimits to A-colimits. For example, let Γ be a graph and F be an A-diagram over Γ. We say that $\mathsf{colim}^A_\Gamma(F)$ is *universal*, or *pullback-stable* [19], if for every pullback square

$$\begin{array}{ccc} \mathsf{colim}^A_\Gamma(F) \times_V Y & \xrightarrow{\pi_2} & Y \\ {\scriptstyle \pi_1}\downarrow & \lrcorner & \downarrow{\scriptstyle h} \\ \mathsf{colim}^A_\Gamma(F) & \xrightarrow[f]{} & V \end{array} \qquad (\mathsf{pb})$$

in $A/\mathcal{U}$, the canonical map

$$\sigma_{f,h} \ : \ \mathsf{colim}^A_{i:\Gamma}(F_i \times_V Y) \to_A \mathsf{colim}^A_\Gamma(F) \times_V Y$$

is an equivalence.[5] The distinguishing feature of a LCC ∞-category, such as $\mathcal{U}$, is that all of its colimits are universal. Although the coslice of a LCC category need not be LCC, we now show that all of its colimits over trees are universal.

▶ **Lemma 19.** *The forgetful functor $\mathcal{F} : A/\mathcal{U} \to A$ preserves limits.*

Proof. The functor $\mathcal{F}$ is right adjoint to the functor $X \mapsto X + A$, so it preserves limits. ◀

▶ **Theorem 20.** *All colimits in $\mathcal{U}$ are universal.*

[5] We show how to construct pullbacks in $A/\mathcal{U}$ in [6, Note 6.0.4].

We have formalized Theorem 20 in Agda (see the folder [7, `Pullback-stability`]).

▶ **Corollary 21.** *For each tree* Γ *and each* A*-diagram* F *over* Γ*, the colimit* $\mathsf{colim}^A_\Gamma(F)$ *is universal.*

Proof. Suppose that Γ is a tree and consider the pullback square (pb). By Theorem 18 combined with Theorem 20, the function

$$\mathsf{ty}(\mathsf{colim}^A_{i:\Gamma}(F_i \times_V Y)) \xrightarrow{\mathsf{fun}(\sigma_{f,h})} \mathsf{ty}(\mathsf{colim}^A_\Gamma(F)) \times_{\mathsf{ty}(V)} \mathsf{ty}(Y)$$

is an equivalence. The codomain is in this form because $\mathcal{F}$ preserves pullbacks by Lemma 19. It follows that $\sigma_{f,h}$ is an equivalence. ◀

7 Preservation of the left class of an OFS

In this section, we combine our construction of $\mathsf{colim}^A_\Gamma(\delta)$ from Section 5.4 with Theorem 13 to prove that colim^A_Γ always preserves the left class of an OFS on $\mathcal{U}$. We assume the univalence axiom to have access to the the tools of univalent bicategories developed in Section 4.

Let $(\mathcal{L}, \mathcal{R})$ be an OFS on $\mathcal{U}$. For all diagrams $F, G : \mathcal{D}_\Gamma :\equiv \mathsf{Diag}(\Gamma, \mathcal{U})$ and natural transformations $(H, \gamma) : F \Rightarrow G$, define the predicates $\widehat{\mathcal{L}}(H, \gamma) :\equiv (i : \Gamma_0) \to \mathcal{L}(H_i)$ and $\widehat{\mathcal{R}}(H, \gamma) :\equiv (i : \Gamma_0) \to \mathcal{R}(H_i)$.

▶ **Lemma 22** ([6, Theorem 7.0.8])**.** *Let* $Q : F \Rightarrow G$*. The following type is contractible:*

$$\mathsf{fact}_{\widehat{\mathcal{L}},\widehat{\mathcal{R}}}(Q) :\equiv \sum_{M:\mathcal{D}_\Gamma} \sum_{S:F \Rightarrow M} \sum_{T:M \Rightarrow G} (T \circ S = Q) \times \widehat{\mathcal{L}}(S) \times \widehat{\mathcal{R}}(T).$$

By Lemma 22, we see that $(\mathcal{L}, \mathcal{R})$ lifts levelwise to $\mathcal{D}_\Gamma$. Since the functor $\mathsf{const}_\Gamma : \mathcal{U} \to \mathcal{D}_\Gamma$ clearly takes $\mathcal{R}$ to $\widehat{\mathcal{R}}$, we deduce that $\mathsf{colim}_\Gamma(-)$ takes $\widehat{\mathcal{L}}$ to $\mathcal{L}$ by Theorem 13.[6]

For each $X, Y : \mathsf{Ob}(A/\mathcal{U})$, consider the predicate $\mathcal{L}_A(f, p) :\equiv \mathcal{L}(f)$ on $X \to_A Y$. Also, define the predicate $\widehat{\mathcal{L}}_A$ on maps of A-diagrams over Γ by $\widehat{\mathcal{L}}_A(H, \gamma) :\equiv \prod_{i:\Gamma_0} \mathcal{L}_A(H_i)$. Then the functor colim^A_Γ takes $\widehat{\mathcal{L}}_A$ to $\mathcal{L}_A$. Indeed, consider a map $\delta : \mathcal{A} \Rightarrow_A \mathcal{B}$ of A-diagrams over Γ. The underlying function of $\mathsf{colim}^A_\Gamma(\delta)$ is induced by the morphism of spans

$$\begin{array}{ccccc} A & \longleftarrow & \mathsf{colim}_\Gamma A & \longrightarrow & \mathsf{colim}_\Gamma(\mathcal{F}(\mathcal{A})) \\ {\scriptstyle \mathsf{id}}\downarrow & & \downarrow{\scriptstyle \mathsf{id}} & & \downarrow{\scriptstyle \bar{\delta}} \\ A & \longleftarrow & \mathsf{colim}_\Gamma A & \longrightarrow & \mathsf{colim}_\Gamma(\mathcal{F}(\mathcal{B})) \end{array}$$

Thus, if δ is in $\widehat{\mathcal{L}}_A$, then all three vertical functions are in $\mathcal{L}$. Since a map of spans is a map of diagrams, we see that $\mathsf{colim}^A_\Gamma(\delta)$ is in $\mathcal{L}_A$.

Recall that a type X is $(\mathcal{L}, \mathcal{R})$-*connected* if the function $X \to \mathbf{1}$ is in $\mathcal{L}$. If F is a diagram of pointed types over Γ such that each $\mathsf{ty}(F_i)$ is $(\mathcal{L}, \mathcal{R})$-connected, then the type $\mathsf{colim}^*_\Gamma(F) :\equiv \mathsf{colim}^{\mathbf{1}}_\Gamma(F)$ is also $(\mathcal{L}, \mathcal{R})$-connected. Indeed, we can deduce that $\mathsf{colim}^*_\Gamma \mathbf{1} = \mathbf{1}$ from the construction of $\mathcal{P}_F$. Thus, colim^*_Γ takes the unique map $F \Rightarrow_* \mathbf{1}$ of pointed diagrams to $(c, c_p) : \mathsf{colim}^*_\Gamma(F) \to_* \mathsf{colim}^*_\Gamma \mathbf{1}$ where $c : \mathsf{colim}^*_\Gamma(F) \to \mathbf{1}$ is the constant map. In addition, $\mathcal{L}(c)$ holds because colim^*_Γ takes $\widehat{\mathcal{L}}_{\mathbf{1}}$ to $\mathcal{L}_{\mathbf{1}}$.

[6] The adjunction $\mathsf{colim}_\Gamma \dashv \mathsf{const}_\Gamma$ follows directly from the equivalence postcomp in Section 3.3.

▶ **Example 23.** Let n be a truncation level [27, Chapter 7]. The archetypal OFS in HoTT has n-connected functions as its left class and n-truncated ones as its right. Thus, by our preceding argument, if each $\mathsf{ty}(F_i)$ is n-connected, then so is $\mathsf{colim}^*_\Gamma(F)$.

Now, let $-1 \leq k < \infty$ also be a truncation level. Recall the category $(n, k)\,\mathsf{GType}$ of *k-tuply groupal n-groupoids* [2]. (These are examples of *higher groups*, in the sense of group theory). This is equivalent to the full subcategory $\mathcal{U}^*_{\geq k, \leq n+k}$ of $\mathcal{U}^*$ on $(k-1)$-connected, $(n+k)$-truncated pointed types. For each truncation level m, consider the m-truncation functor $\|-\|_m : A/\mathcal{U} \to A/\mathcal{U}$, which takes a type X to the universal m-type admitting a function from X [27, Section 7.3]. This functor preserves colimits as a left adjoint [6, Proposition 3.4.6], and its associated counit is an isomorphism. It follows that $\mathcal{U}^*_{\geq k, \leq n+k}$, hence $(n, k)\,\mathsf{GType}$, inherits colimits over graphs from $\mathcal{U}^*$.

It is a special feature of *pointed* colimits that they always preserve n-connectedness. If Γ is not a tree, then colim_Γ may fail to preserve n-connectedness. Indeed, let Γ be the graph with a single point $*$ and a single edge from $*$ to itself. Define the diagram F over Γ by $F_0(*) :\equiv \mathbf{1}$ and $F_{*,*,*} :\equiv \mathsf{id}_{\mathbf{1}}$. Then $\mathsf{colim}_\Gamma(F)$ is equivalent to the circle S^1, which is not 1-connected.

▶ Remark 24. Although Example 23 is only about pointed types, we do benefit from the formulation of the main connection for general coslices. Indeed, for each object G of $\mathcal{U}^*_{\geq k, \leq n+k}$, the coslice $G/\mathcal{U}^*_{\geq k, \leq n+k}$ is a reflective subcategory of $\mathsf{ty}(G)/\mathcal{U}_{\geq k, \leq n+k}$ for which the reflector is 2-*coherent* [6, Definition B.0.1]. (Here $\mathcal{U}_{\geq k, \leq n+k}$ denotes the subuniverse of $(k-1)$-connected, $(n+k)$-truncated types.) Hence $G/\mathcal{U}^*_{\geq k, \leq n+k}$ inherits colimits over graphs from the coslice $\mathsf{ty}(G)/\mathcal{U}_{\geq k, \leq n+k}$ [6, Corollary 7.1.3]. In its full generality, Section 5 gives us an explciit construction of such colimits.

In particular, let $n, m : \mathbb{N}$ with $n > 0$ and $m < n$. The Eilenberg-MacLane space $K(\mathbb{Z}, n)$ [13] is the free group on one generator in the category $(n, m)\,\mathsf{GType}$. Therefore, when $m > 0$, we may view $K(\mathbb{Z}, n)/\mathcal{U}^*_{\geq m, \leq n+m}$ as a higher version of the category of *pointed abelian groups* [18]. We see, then, that Section 5 lets us build colimits of higher pointed abelian groups inside HoTT.

8 Mapping colimits to weak limits

Finally, we look at the interaction between colimits and (Eilenberg-Steenrod) cohomology theories. Specifically, we apply the 3×3 lemma to the main connection to obtain the familiar construction of $\mathsf{colim}^A_\Gamma(F)$ as a pushout of coproducts in $A/\mathcal{U}$. Afterward, we apply this new construction to the Mayer-Vietoris sequence to prove that cohomology theories send finite colimits to weak limits in **Set** assuming the axiom of choice.

8.1 Decomposition of A-colimits into simpler pieces

To make use of the 3×3 lemma, we first form the following grid of commuting squares:

$$
\begin{array}{ccccc}
\sum_{i,j,g} \mathsf{ty}(F_i) & \xleftarrow{\mathsf{id}+\mathsf{id}} & \left(\sum_{i,j,g} \mathsf{ty}(F_i)\right) + \left(\sum_{i,j,g} \mathsf{ty}(F_i)\right) & \xrightarrow{(i,x)+(j,\mathsf{fun}(F_{i,j,g})(x))} & \sum_i \mathsf{ty}(F_i) \\
\big\uparrow{\scriptstyle (i,j,g,\mathsf{str}(F_i)(a))} & & \big\uparrow{\scriptstyle (i,j,g,\mathsf{str}(F_i)(a))+(i,j,g,\mathsf{str}(F_i)(a))} & & \big\uparrow{\scriptstyle (i,\mathsf{str}(F_i)(a))} \\
\left(\sum_{i,j} \Gamma_1(i,j)\right) \times A & \xleftarrow{\mathsf{id}+\mathsf{id}} & \left(\left(\sum_{i,j} \Gamma_1(i,j)\right) \times A\right) + \left(\left(\sum_{i,j} \Gamma_1(i,j)\right) \times A\right) & \xrightarrow{(i,a)+(j,a)} & \Gamma_0 \times A \\
\big\downarrow{\scriptstyle \mathsf{pr}_2} & & \big\downarrow{\scriptstyle \mathsf{pr}_2+\mathsf{pr}_2} & & \big\downarrow{\scriptstyle \mathsf{pr}_2} \\
A & \xleftarrow[\mathsf{id}_A]{} & A & \xrightarrow[\mathsf{id}_A]{} & A
\end{array}
$$

Call the pushouts of the left, middle, and right vertical spans V_1, V_2, and V_3, respectively. Call the pushouts of the top, middle, and bottom horizontal spans H_1, H_2, and H_3, respectively. We can form two additional pushouts from this grid:

$$\begin{array}{ccc} V_2 & \xrightarrow{\delta_2} & V_3 \\ {\scriptstyle \delta_1}\downarrow & & \downarrow{\scriptstyle \mathsf{inr}} \\ V_1 & \xrightarrow[\mathsf{inl}]{} & P_V \end{array} \qquad \begin{array}{ccc} H_2 & \xrightarrow{\eta_1} & H_1 \\ {\scriptstyle \eta_2}\downarrow & & \downarrow{\scriptstyle \mathsf{inr}} \\ H_3 & \xrightarrow[\mathsf{inl}]{} & P_H \end{array}$$

- δ_1 denotes the function induced by the middle-to-left map of spans;
- δ_2 denotes the function induced by the middle-to-right map of spans;
- η_1 denotes the function induced by the middle-to-top map of spans; and
- η_2 denotes the function induced by the middle-to-bottom map of spans.

The 3×3 lemma now gives us an equivalence $\tau_1 : P_H \xrightarrow{\simeq} P_V$ of types defined by double induction on pushouts [12, Section VII].

▶ Note. Let Δ be a discrete graph and G an A-diagram over Δ. The pushout

$$\begin{array}{ccc} \Delta_0 \times A & \xrightarrow{(i,a)\mapsto(i,\mathsf{str}(G_i)(a))} & \sum_{i:\Delta_0} \mathsf{ty}(G_i) \\ {\scriptstyle \mathsf{pr}_2}\downarrow & & \downarrow{\scriptstyle \mathsf{inr}} \\ A & \xrightarrow[\mathsf{inl}]{} & D \end{array}$$

together with inl is the coproduct of the G_i in $A/\mathcal{U}$. We denote D by $\bigvee_{i:\Delta_0} \mathsf{ty}(G_i)$.

▶ **Lemma 25.** *We have two equivalences of spans*

$$\begin{array}{ccccc} A & \xleftarrow{[\mathsf{id}_A]} & \mathsf{colim}_\Gamma A & \xrightarrow{\psi} & \mathsf{colim}_\Gamma(\mathcal{F}(F)) \\ \simeq\downarrow & & \simeq\downarrow & & \simeq\downarrow \\ H_3 & \xleftarrow[\eta_2]{} & H_2 & \xrightarrow[\eta_1]{} & H_1 \end{array}$$

$$\begin{array}{ccccc} V_1 & \xleftarrow{\delta_1} & V_2 & \xrightarrow{\delta_2} & V_3 \\ \Vert & & \simeq\downarrow & & \Vert \\ \bigvee_{i,j,g} \mathsf{ty}(F_i) & \xleftarrow[\mathsf{id}\vee\mathsf{id}]{} & \left(\bigvee_{i,j,g} \mathsf{ty}(F_i)\right) \vee \left(\bigvee_{i,j,g} \mathsf{ty}(F_i)\right) & \xrightarrow[\nu]{} & \bigvee_i \mathsf{ty}(F_i) \end{array}$$

where ν is defined by double induction on pushouts through the commuting square

$$\begin{array}{ccc} A & \longrightarrow & \bigvee_{i,j,g} \mathsf{ty}(F_i) \\ \downarrow & \mathsf{refl}_{\mathsf{inl}(a)} & \downarrow{\scriptstyle (i,j,g,x)\mapsto\mathsf{inr}(j,\mathsf{fun}(F_{i,j,g})(x))} \\ \bigvee_{i,j,g} \mathsf{ty}(F_i) & \xrightarrow[(i,j,g,x)\mapsto\mathsf{inr}(i,x)]{} & \bigvee_i \mathsf{ty}(F_i) \end{array}$$

Notice that the pushout of the topmost span appearing in Lemma 25 is exactly $\mathcal{P}_F$. As a result, the equivalence supplied by the 3×3 lemma gives us $\mathsf{colim}^A_\Gamma(F)$ as a familiar pushout of coproducts.

▶ **Corollary 26** ([6, Corollary 5.5.3]). *We have a pushout square*

$$\begin{array}{ccc}
\left(\bigvee_{i,j,g} \mathsf{ty}(F_i)\right) \vee \left(\bigvee_{i,j,g} \mathsf{ty}(F_i)\right) & \xrightarrow{\nu} & \bigvee_i \mathsf{ty}(F_i) \\
\Big\downarrow{\scriptstyle \mathsf{id} \vee \mathsf{id}} & & \Big\downarrow \\
\bigvee_{i,j,g} \mathsf{ty}(F_i) & \longrightarrow & \mathsf{colim}^A_\Gamma(F)
\end{array}$$

8.2 Weak continuity of cohomology

With this new construction of colim^A_Γ, we can transfer the weak continuity of cohomology to HoTT. This application is described in detail in [6, Section 8].

Let $\mathbf{Ab}$ denote the category of abelian groups. Suppose that $H^* : (\mathcal{U}^*)^{\mathrm{op}} \to \mathbf{Ab}$ is a cohomology theory.[7] Consider the pushout

$$\begin{array}{ccc}
Z & \xrightarrow{g} & Y \\
{\scriptstyle f}\Big\downarrow & & \Big\downarrow{\scriptstyle \mathsf{inr}} \\
X & \xrightarrow[\mathsf{inl}]{} & P
\end{array}$$

of a span of pointed maps. In [4], Cavallo constructs the *Mayer-Vietoris sequence* for P, a long exact sequence (LES) of the form

$$\cdots \quad H^{n-1}(P) \longrightarrow H^{n-1}(X) \times H^{n-1}(Y) \longrightarrow H^{n-1}(Z)$$

$$\longrightarrow H^n(P) \xrightarrow[(H^n(\mathsf{inl}),H^n(\mathsf{inr}))]{} H^n(X) \times H^n(Y) \xrightarrow[H^n(f)-H^n(g)]{} H^n(Z) \quad \cdots$$

Let F be a diagram of pointed types over a *finite* graph Γ, which means that Γ_0 is finite and $\Gamma_1(i,j)$ is finite for all $i, j : \Gamma_0$. As cohomology preserves finite wedges [4, Section 4.2], Corollary 26 combined with this LES gives us an exact squence

$$H^n(\mathsf{colim}^*_\Gamma(F)) \xrightarrow{\zeta_n} \prod_{i,j,g} H^n(F_i) \times \prod_i H^n(F_i) \xrightarrow{\mu_n - \nu_n} \prod_{i,j,g} H^n(F_i) \times \prod_{i,j,g} H^n(F_i) \tag{prod-cohom}$$

for each $n : \mathbb{Z}$.[8] Here, ζ_n is defined as the composite

$$\begin{array}{ccc}
H^n(\mathsf{colim}^*_\Gamma(F)) & \overset{\zeta_n}{\dashrightarrow} & \prod_{i,j,g} H^n(F_i) \times \prod_i H^n(F_i) \\
\Big\downarrow{\scriptstyle (H^n(\mathsf{inl}),H^n(\mathsf{inr}))} & \nearrow{\scriptstyle \cong \times \cong} & \\
H^n(\bigvee_{i,j,g} F_i) \times H^n(\bigvee_i F_i) & &
\end{array}$$

and μ_n and ν_n are defined by $(f, h) \mapsto (f, \lambda i \lambda j \lambda g. H^n(F_{i,j,g})(h_j))$ and $(f, h) \mapsto (f, \lambda i \lambda j \lambda g. h_i)$, respectively. Moreover, the universal property of limits in $\mathbf{Ab}$ induces a commuting triangle

[7] See [1, Section 6] for a description of Eilenberg-Steenrod cohomology theory inside HoTT. A slightly more general definition, which works in our setting, is found in [6, Section 8.1].

[8] When H^* is a singular cohomology theory, we may extend the class of graphs to those satisfying the set-level axiom of choice, in the sense of [1, Definition 6.1].

$$\begin{array}{ccccc} H^n(\mathsf{colim}^*_\Gamma(F)) & & \xrightarrow{\Delta_F} & & \mathsf{lim}_\Gamma\, H^n(F) \\ & \searrow^{H^n(\iota_i)} & & {}^{\mathsf{pr}_i}\swarrow & \\ & & H^n(F_i) & & \end{array}$$

for each $i : \Gamma_0$, induced by the cone $(H^n(\mathsf{colim}^*_\Gamma(F)), H^n(\iota))$ over $H^n(F)$. One can check that the exactness of (`prod-cohom`) implies that Δ_F is epic as a map of sets.

At this stage, if we were in a classical system, then it would follow that Δ_F has a section, which in turn would imply that $H^n(\mathsf{colim}^*_\Gamma(F))$ is a weak limit in **Set**. Inside HoTT, we may assume the axiom of choice [27, Chapter 3.8] to conclude that Δ_F is *merely* a weak limit.[9] In this sense, H^* enjoys a restricted version of weak continuity inside HoTT.

9 Conclusion and future work

We explored colimits inside HoTT. The heart of our work was the connection between A-colimits and ordinary colimits, i.e., *the main connection*. To derive the main connection, we found an explicit construction of A-colimits that was tailored to reveal the connection. We used the main connection to prove that the forgetful functor from a coslice creates colimits over trees and that A-colimits over trees are universal. We also used the main connection to examine how colimits interact with factorization systems. As a result, we found that all pointed colimits preserve n-connectedness, which implies that higher groups are closed under colimits on directed graphs. Finally, we used the main connection to see that cohomology takes finite colimits to weak limits in **Set** assuming the axiom of choice.

A natural direction is to extend our development to colimits of diagrams over 2-computads [26]. To our knowledge, colimits of type-valued diagrams over higer-dimensional graphs have not been developed in the untruncated setting. We believe both Section 6 and Section 7 can be generalized to the setting of 2-computads.

References

1. Ulrik Buchholtz and Kuen-Bang Hou (Favonia). Cellular Cohomology in Homotopy Type Theory. *Logical Methods in Computer Science*, Volume 16, Issue 2, 2020. `doi:10.23638/LMCS-16(2:7)2020`.
2. Ulrik Buchholtz, Floris van Doorn, and Egbert Rijke. Higher Groups in Homotopy Type Theory. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '18, pages 205–214, New York, NY, USA, 2018. Association for Computing Machinery. `doi:10.1145/3209108.3209150`.
3. Paolo Capriotti and Nicolai Kraus. Univalent higher categories via complete Semi-Segal types. *Proc. ACM Program. Lang.*, 2(POPL), 2017. `doi:10.1145/3158132`.
4. Evan Cavallo. Synthetic Cohomology in Homotopy Type Theory. Master's thesis, Carnegie Mellon University, 2015. URL: `https://staff.math.su.se/evan.cavallo/works/thesis15.pdf`.
5. J Daniel Christensen and Luis Scoccola. The Hurewicz theorem in homotopy type theory. *Algebraic and Geometric Topology*, 23(5):2107–2140, 2023. `doi:10.2140/agt.2023.23.2107`.
6. Perry Hart and Kuen-Bang Hou. Coslice Colimits in Homotopy Type Theory, 2024. `arXiv:2411.15103`.
7. Perry Hart and Kuen-Bang Hou (Favonia). A formalized construction of coslice colimits, 2024. v0.1.0. URL: `https://github.com/PHart3/colimits-agda/tree/v0.1.0`.

[9] As in [27, Chapter 3.10], the adverb *merely* refers to propositional truncation.

8 Kuen-Bang Hou (Favonia), Eric Finster, Daniel R. Licata, and Peter LeFanu Lumsdaine. A Mechanization of the Blakers-Massey Connectivity Theorem in Homotopy Type Theory. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '16, pages 565–574, New York, NY, USA, 2016. Association for Computing Machinery. doi:10.1145/2933575.2934545.

9 Krzysztof Kapulkin and Peter LeFanu Lumsdaine. The simplicial model of Univalent Foundations (after Voevodsky). *J. Eur. Math. Soc.*, 23(6):2071–2126, 2021. doi:10.4171/JEMS/1050.

10 Max Kelly. On MacLane's Conditions for Coherence of Natural Associativities, Commutativities, etc. *Journal of Algebra*, 1(4):397–402, 1964. doi:10.1016/0021-8693(64)90018-3.

11 Nicolai Kraus and Jakob von Raumer. Path Spaces of Higher Inductive Types in Homotopy Type Theory . In *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–13, Los Alamitos, CA, USA, June 2019. IEEE Computer Society. doi:10.1109/LICS.2019.8785661.

12 Daniel R. Licata and Guillaume Brunerie. A Cubical Approach to Synthetic Homotopy Theory. In *2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 92–103, 2015. doi:10.1109/LICS.2015.19.

13 Daniel R. Licata and Eric Finster. Eilenberg-MacLane spaces in homotopy type theory. In *Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, CSL-LICS '14, New York, NY, USA, 2014. Association for Computing Machinery. doi:10.1145/2603088.2603153.

14 Jacob Lurie. Higher Algebra. Unpublished, 2017. URL: https://www.math.ias.edu/~lurie/papers/HA.pdf.

15 Jacob Lurie. Kerodon. https://kerodon.net, 2024.

16 nLab authors. created limit. https://ncatlab.org/nlab/show/created+limit, 2024. Revision 21.

17 nLab authors. (infinity,1)-limit. https://ncatlab.org/nlab/show/%28%E2%88%9E%2C1%29-limit, 2024. Revision 78.

18 nLab authors. pointed abelian group. https://ncatlab.org/nlab/show/pointed+abelian+group, November 2024. Revision 3.

19 nLab authors. pullback-stable colimit. https://ncatlab.org/nlab/show/pullback-stable+colimit, October 2024. Revision 18.

20 Emily Riehl. *Categorical Homotopy Theory*. New Mathematical Monographs. Cambridge University Press, 2014. doi:10.1017/CBO9781107261457.

21 Egbert Rijke. Introduction to Homotopy Type Theory, 2022. arXiv:2212.11082.

22 Egbert Rijke, Michael Shulman, and Bas Spitters. Modalities in homotopy type theory. *Logical Methods in Computer Science*, Volume 16, Issue 1, January 2020. doi:10.23638/LMCS-16(1:2)2020.

23 Egbert Rijke, Elisabeth Stenholm, Jonathan Prieto-Cubides, Fredrik Bakke, and others. The agda-unimath library. URL: https://github.com/UniMath/agda-unimath/.

24 Kristina Sojakova. Higher Inductive Types as Homotopy-Initial Algebras. *SIGPLAN Not.*, 50(1):31–42, 2015. doi:10.1145/2775051.2676983.

25 Kristina Sojakova, Floris van Doorn, and Egbert Rijke. Sequential Colimits in Homotopy Type Theory. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '20, pages 845–858, 2020. doi:10.1145/3373718.3394801.

26 Ross Street. Limits indexed by category-valued 2-functors. *Journal of Pure and Applied Algebra*, 8(2):149–181, 1976. doi:10.1016/0022-4049(76)90013-X.

27 The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. https://homotopytypetheory.org/book, Institute for Advanced Study, 2013.

28 Floris van Doorn, Jakob von Raumer, and Ulrik Buchholtz. Homotopy Type Theory in Lean. In *Interactive Theorem Proving*, pages 479–495. Springer International Publishing, 2017. doi:10.1007/978-3-319-66107-0_30.