Classifying 2-Groups in Homotopy Type Theory

Perry Hart
hart1262@umn.edu
University of Minnesota
Minneapolis, Minnesota, USA

Owen Milner
omilner@andrew.cmu.edu
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA

Abstract

Under the homotopy hypothesis, higher dimensional groups are defined as pointed homotopy types whose homotopy groups vanish outside a certain range. In particular, a 2-*group* is a pointed connected homotopy 2-type. Classically, 2-groups have two equivalent algebraic descriptions: one in terms of weak monoidal categories and the other in terms of group cohomology. We present these two classifications of pointed connected 2-types in homotopy type theory, thereby providing internal, constructive counterparts to the traditional classifications of 2-groups. Our first classification (in terms of monoidal categories) takes the form of a bicategorical equivalence, while our second is a type equivalence that extends to n-groups for all integers $n \geq 2$. We have mechanized our results in Agda.

CCS Concepts

• **Theory of computation** → **Type theory**; *Constructive mathematics*; *Logic and verification*; • **Mathematics of computing** → **Algebraic topology**.

Keywords

homotopy type theory, synthetic homotopy theory, 2-group, higher inductive type, bicategory, higher group, cohomology

1 Introduction

Groups are fundamental to modern algebra. By treating a group's equational laws as isomorphisms in a category so that multiplication is a monoidal product and inverses are adjoint equivalences, we arrive at a *coherent* 2-*group*, the 2-dimensional generalization of a group. The study of coherent 2-groups has a fruitful history spanning algebraic topology and mathematical physics [4, 5, 23]. From the perspective of homotopy theory, coherent 2-groups fit into a layered correspondence, known as the *homotopy hypothesis*, between spaces and higher dimensional groupoids [3]. The homotopy hypothesis provides a uniform definition of higher groups in homotopy type theory (HoTT) [7]. Following [20], we call this definition the *internal* notion of higher group. It lets us apply the tools of synthetic homotopy theory directly to group theory at any dimension. In dimension one, this application has led to simpler, fundamentally new proofs of important group-theoretic results [7, 15, 27], such as the Nielsen-Schreier theorem. The application in dimension two is just beginning [28].

This paper makes two principal contributions—which we discuss now—to higher group theory in HoTT, with each shedding new light on the internal notion of higher group.

The homotopy hypothesis asserts that this internal notion (as a space) is equivalent to the *axiomatic* notion (as a groupoid). In the 1-dimensional case, the latter is the usual set-theoretic definition of a (possibly abelian) group. In this case, we can prove the homotopy hypothesis in HoTT. In particular, applying the loop space to a pointed connected 1-type—the internal notion of a 1-group—forms an equivalence between the 1-category of such types and the 1-category of axiomatic groups [7, Theorem 5.1], with the functor in the reverse direction forming the *delooping* of an axiomatic group. This equivalence is useful for group theory in HoTT as we can integrate two kinds of proofs: those about internal groups developed with synthetic homotopy theory and those about axiomatic groups developed in the set-theoretic setting.

It's natural to ask for a similar equivalence between the internal and axiomatic notion above dimension one. We have lacked one, however, as the higher structure makes even the 2-dimensional case far harder. Buchholtz, van Doorn, and Rijke conjecture that the 2-dimensional case is provable in HoTT [7, Section 9]. In classical homotopy theory, Baez and Lauda conjecture the *basic* 2-dimensional case, i.e., for the spaces sitting at the first connectivity level [4, Section 8.2].

Our first contribution settles the basic 2-dimensional case in HoTT: Working in HoTT [29], we show that applying the loop space to a pointed connected 2-type (the internal notion of 2-group) forms a *(biadjoint) biequivalence* [11, Definition 2.3] between the $(2, 1)$-category of such types and the $(2, 1)$-category of coherent 2-groups (the axiomatic notion of 2-group). To this end, we formulate a notion of biequivalence that is simpler than the traditional one but still, for *univalent* bicategories, equivalent to identity. As a result, the biequivalence we build is equivalent to a path between these $(2, 1)$-categories, so that one can easily transfer any bicategorical property between them.

The biequivalence we build extends the preceding equivalence constructed for 1-groups. It has two main stages. First,

Conference'17, Washington, DC, USA
2026. ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

we form the delooping of a coherent 2-group G as a higher inductive type that generalizes the delooping of a group due to Licata and Finster [17]. Intuitively, the delooping is the smallest pointed 2-type T admitting a 2-group morphism from G to T's loop space $\Omega(T)$, which turns out to be an isomorphism by minimality. The delooping defines a map from the type of coherent 2-groups to that of pointed connected 2-types. Second, we make this map into a *pseudofunctor* (a bicategorical functor) and prove that it forms a biequivalence with the loop pseudofunctor Ω.

We now see that the loop space $\Omega(X)$ of a pointed connected 2-type X preserves all information in a bicategorical sense. What if we only have access to the fundamental group $\pi_1(X) :\equiv \|\Omega(X)\|_0$ of X? Unsurprisingly, we need more information to classify X, but how much more? In classical homotopy theory, MacLane and Whitehead showed X is determined by $\pi_1(X)$, the canonical action of $\pi_1(X)$ on $\pi_2(X)$, and a group cohomology class in $H^3(\pi_1(X), \pi_2(X))$ called the *Postnikov invariant* of X [19]. These three pieces of data make up a *Sính triple* (named after Hoàng Xuân Sính), which consists of a group G, a G-module H, and a cohomology class $\kappa \in H^3(G, H)$. MacLane and Whitehead's classification arises from a set-theoretic bijection—which we call the *MW bijection*—between the pointed connected 2-types and the isomorphism classes of Sính triples. In HoTT, the pointed connected 2-types form not a set but a 2-type. How, then, can we turn the MW bijection into a type equivalence?

Our second contribution answers this question with an equivalence between the type of pointed connected 2-types and the type of *untruncated Sính triples*—triples as before but with κ as a cocycle (an element of the relevant untruncated mapping space) rather than cohomology class (which lives in the set-truncated space). Moreover, by taking the set truncation of this equivalence, we recover the MW bijection—here between the (connected) components (equivalently, *mere* isomorphism classes) of pointed connected 2-types and the components of Sính triples. We thus recover the Postnikov invariant of a pointed connected 2-type. Our type equivalence stems from general results about deloopings of types and so yields a substantially different proof of the MW bijection from MacLane and Whitehead's. In particular, ours is constructive and works in dimension two and above.

Indeed, our equivalence is uniform in dimension n, as one between the pointed connected $(n+1)$-types—i.e., internal $(n+1)$-groups—and the *untruncated Sính n-triples* for $n \geq 1$. It sends an internal $(n+1)$-group X to the triple consisting of X's fundamental internal n-group $\Pi_n(X)$, the canonical action of $\Pi_n(X)$ on $\pi_{n+1}(X)$, and an invariant of X in the form of an $(n+2)$-dimensional cocycle on $\Pi_n(X)$ over $\pi_{n+1}(X)$.

With our first classification of pointed connected 2-types (from the biequivalence), we get a bijection between the components of coherent 2-groups and those of Sính triples. The Postnikov invariant produced for a 2-group G by this composite equivalence is traditionally called the *Sính invariant* of G, and ours is the first construction of it in type theory.

Outline. We consider our two contributions in turn, starting with the biequivalence. We first review basic notions of bicategory theory while focusing on the $(2, 1)$-category of pointed connected 2-types and that of coherent 2-groups (Section 4). Afterward, we outline the computations involved in the two stages of the biequivalence between them (Sections 5 and 6). We deduce from the biequivalence an identity between the two $(2, 1)$-categories via univalence (Section 7).

For our second contribution, we begin by reviewing some key results on deloopings of types (Section 8). With these results, we construct, for each $n \geq 1$, an equivalence between the type of internal $(n+1)$-groups and that of untruncated Sính n-triples (Section 9). We then derive a bijection between the components of internal $(n+1)$-groups and the components of Sính n-triples, where the cocycle is replaced by a cohomology class, as in the MW bijection.

Agda formalization. This paper can serve as a roadmap for our Agda codebase [12], which is completely self-contained and formalizes our entire development. Hyperlinks to the code will be blue and in brackets. Besides our new results, the codebase includes important theorems from [7, 32], offering the first, to our knowledge, Agda formalization of [7]'s higher delooping theorem for abelian groups.

To check the biequivalence, Agda needs a lot of time and memory despite our careful engineering of the code. Part of this high computational cost highlights a major difference between our type system—Book HoTT—and cubical type theory [2, 31]: Cubical has *definitional β*-rules for path constructors in higher inductive types (HITs), which greatly simplifies the biequivalence by erasing many postulated equalities that we must handle. Although constructions with HITs tend to be much harder in Book HoTT [21], they have value in this setting. Book HoTT has models in all $(\infty, 1)$-toposes [18, 26], whereas it's not known whether the type theory underlying Cubical Agda has a model Quillen equivalent to the category of spaces. Moreover, cubical is an extension of Book HoTT [2, Section 2.16], so we can interpret our results into it.

2 Related work

Baez and Lauda introduced the notion of a coherent 2-group in the language of classical category theory [4, Section 3]. In HoTT, Veltri and van der Weide internalized its definition as an example of an algebra over a signature [30, Section 7.4] and thereby proved the bicategory of coherent 2-groups is univalent (of which we give a direct proof in Section 4). Baez and Lauda gave—without passing through spaces—a modern proof that coherent 2-groups are in bijection with Sính triples [4, Corollary 8.3.8], a result they trace to Sính's PhD thesis. Unlike ours, their proof is non-constructive. They take the skeleton $\mathcal{S}$ of an arbitrary coherent 2-group's underlying category, and the existence of $\mathcal{S}$ requires choice [14].

Noohi provides a different but closely related classical classification of pointed connected 2-types [23, Proposition 6.1]. He considers the category **2Gp** of strict 2-groups with weak equivalences those maps $G_1 \to G_2$ inducing isomorphisms on the group of isomorphism classes $\pi_0(-)$ and the automorphism group $\mathrm{Aut}(e_{(-)})$. He proves the nerve of a 2-category $N : \mathbf{2Cat} \to \mathbf{sSet}$ induces a 1-categorical equivalence between $\mathrm{Ho}(\mathbf{2Gp})$ and the pointed connected 2-types.

Additionally, Baez and Lauda prove that strict 2-groups are classified by *crossed modules*, which are pairs of related groups—a concept that also is central to MacLane and Whitehead's proof. In HoTT, Buchholtz and Schipp von Branitz show that the type of strict 2-groups is equivalent to that of crossed modules [8]. They also conjecture that, with the *sets cover 1-types* axiom, coherent 2-groups can be strictified.

3 Background on type theory

We assume the reader is familiar with HoTT as in [29], in which our work takes place. This system extends Martin-Löf type theory with the univalence axiom and HITs. For convenience, we review a few basic constructions in HoTT. But first, a remark on notation: We use $(a : A) \to B(a)$ for the type $\prod_{a:A} B(a)$ and $(a : A) \times B(a)$ for the type $\sum_{a:A} B(a)$.

The first construction is the function $\mathsf{ap}_f : (x = y) \to (f(x) = f(y))$ defined by path induction for all functions $f : X \to Y$ and $x, y : X$. (We use $=$ for the identity/path type and $\equiv$ for definitional equality.)

Lemma 3.1 (Homotopy naturality). *Let $f, g : X \to Y$. For all $x, y : X$, $p : x = y$, and $H : f \sim g$, the square*

$$\begin{array}{ccc} f(x) & \overset{H(x)}{=\!=\!=} & g(x) \\ \mathsf{ap}_f(p) \Vert & & \Vert \mathsf{ap}_g(p) \\ f(y) & \underset{H(y)}{=\!=\!=} & g(y) \end{array}$$

commutes. We denote the path witnessing that this square commutes by $\mathsf{hnat}_H(p)$, but for brevity we may hide H and refer to the path as homotopy naturality at p.

Here, $f_1 \sim f_2 :\equiv (x : X) \to f_1(x) = f_2(x)$ for any $f_1, f_2 : (x : X) \to Y(x)$, called the type of *homotopies* between f_1 and f_2. We compose homotopies $H_1 * H_2$ by pointwise path composition. Homotopy naturality respects composition.

Lemma 3.2. *Let $f, g, h : X \to Y$, $H_1 : f \sim g$, and $H_2 : g \sim h$. For all $x, y : X$ and $p : x = y$, we have a path*

$$\begin{array}{ccc} f(x) & = & h(x) \\ \Vert & \mathsf{hnat}_{H_1 * H_2}(p) & \Vert \\ f(y) & = & h(y) \end{array} \quad = \quad \begin{array}{ccc} f(x) & = & g(x) \\ \Vert & \mathsf{hnat}_{H_1}(p) & \Vert \\ f(y) & = & g(y) \end{array} \cdot \begin{array}{ccc} g(x) & = & h(x) \\ \Vert & \mathsf{hnat}_{H_2}(p) & \Vert \\ g(y) & = & h(y) \end{array}$$

where $\cdot$ denotes horizontal composition of squares.

The second construction is the *transport* function $\mathsf{transp}^Y : (x, y : X) \to (x = y) \to Y(x) \to Y(y)$ for any type family Y over X. This is defined by path induction.

The final construction is the *n-truncation* $\|-\|_n$ HIT for each integer $n \geq -2$, an operation on types. (When $n = -1$, we write $\|-\|$.) For any type X, we have a function $|-|_n : X \to \|X\|_n$. Further, $\|X\|_n$ is *n-truncated*, or an *n-type*, a notion defined recursively: a type is -2*-truncated* if it's contractible (i.e., equivalent to the unit type $\mathbf{1}$) and $(n+1)$*-truncated* if all of its identity types are n-truncated. We call the unique element of a contractible type its *center*. We call -1-types *propositions* and 0-types *sets*. For example, the type (X *is an n-type*) is a proposition. Finally, X is *n-connected* if $\|X\|_n$ is contractible. It is *connected* if it's 0-connected. We call $\|X\|_0$ the type of *(connected) components* of X.

4 Bicategories

To give our first classification of internal 2-groups, we need to discuss *bicategories*. For us, *bicategory* means $(2, 1)$-category whose 2-cells (maps between maps) are paths. This definition is a special case of the standard one, in which 2-cells are just assumed to be elements of a family of sets [1, Definition 2.1].

Definition 4.1 (BicatStr). A *bicategory* consists of a type Ob of objects and

- a family hom of 1-types twice indexed over Ob, whose elements are called *morphisms*, *maps*, or 1*-cells*
- a composition operation $\circ : \mathsf{hom}(b, c) \to \mathsf{hom}(a, b) \to \mathsf{hom}(a, c)$ for all $a, b, c : \mathsf{Ob}$
- an identity map id_a for each $a : \mathsf{Ob}$ along with a *right unitor* and a *left unitor*: two 2-cells witnessing that id_a is a right unit and a left unit, respectively, for $\circ$
- an *associator* 2-cell witnessing that $\circ$ is associative and satisfying the triangle and pentagon identities.

Remark 4.2. A bicategory as in Definition 4.1 is equivalent to a *locally univalent bicategory* in the sense of [1, Definition 3.1] all of whose 2-cells are invertible.

Definition 4.3 (Adjequiv). Let C be a bicategory. Let $a, b : \mathsf{Ob}(C)$ and $f : \mathsf{hom}_C(a, b)$. We say that f is an *adjoint equivalence* if we have a morphism $g : \mathsf{hom}_C(b, a)$, 2-cells $\eta : \mathsf{id}_a = g \circ f$ and $\epsilon : f \circ g = \mathsf{id}_b$, and two zig-zag identities. We denote the type of adjoint equivalences by $\mathsf{AdjEquiv}$.

Note that the data of an adjoint equivalence on a morphism is a proposition [Adjequiv-is-prop], not extra structure.

Example 4.4. We have the bicategory $\mathbf{2Type}_0^*$ of pointed connected 2-types and pointed maps [Ptd-bc]. Its hom-types are 1-truncated by [7, Corollary 4.3].

Example 4.5. We have the bicategory **c2Grp** of *(coherent)* 2-*groups* [2Grp-bc]. A 2-group is defined as a (univalent) monoidal groupoid where, from the viewpoint of a monoidal groupoid as a single-object bicategory, every object is an adjoint equivalence. Explicitly, given a universe $\mathcal{U}$, a 2-*group (in* $\mathcal{U}$*)* [CohGrp] is a 1-type G in $\mathcal{U}$ along with

- a basepoint id, called the *unit* of G
- an operation $\otimes : G \to G \to G$, called the *tensor product* of G
- a right unitor ρ, left unitor λ, and associator α that together satisfy the triangle and pentagon identities
- an *inverse* operation $(-)^{-1} : G \to G$
- families of paths $\mathsf{linv}_x : x^{-1} \otimes x = \mathsf{id}$ and $\mathsf{rinv}_x : x \otimes x^{-1} = \mathsf{id}$ related by the zig-zag identities.

A 2-*group morphism* $G_1 \to G_2$ is a function $f_0 : G_1 \to G_2$ equipped with a family of paths $\mu_{x,y} : f_0(x \otimes y) = f_0(x) \otimes f_0(y)$ that respects the associator [CohGrpHomStr].

Note 4.6. Our notion of 2-group morphism is surprisingly short: a coherent map of the underlying semigroups. The correct notion must ensure that the map of underlying types preserves *all* the 2-group data [CohGrpHomStrFull]. To justify our short definition, we prove that for each map $f_0 : G_1 \to G_2$ of the underlying types of 2-groups, the forgetful function

$$\textit{fully explicit notion on } f_0 \quad \to \quad \textit{short notion on } f_0 \qquad (\partial)$$

is an equivalence [2GrpHomEq]. We outline the proof in Section A. The short definition is highly valuable (especially for the formalization) as it lets us define the delooping of a 2-group G as a HIT $\mathcal{K}_2(G)$ with fewer constructors (Section 5), thereby making induction on $\mathcal{K}_2(G)$ far simpler.

The structure identity principle (SIP) [24, Theorem 11.6.2]—a general theorem characterizing identity types of Σ-types—implies that a 2-cell between 2-group morphisms $f, g : G_1 \to G_2$ is equivalent to a *(monoidal) natural isomorphism (or iso)* between them: a homotopy $\mathsf{fun}(f) \sim \mathsf{fun}(g)$ of the underlying functions with a proof it commutes with the tensor product. For example, we build the unitors and associator for **c2Grp** via natural isomorphisms [2SGrpMap].

Example 4.7 (Hmtpy2Grp). For every pointed 2-type X, the loop space $\Omega(X) :\equiv (\mathsf{pt}_X = \mathsf{pt}_X)$ equipped with path composition is a 2-group, called the *fundamental* 2-*group* of X. For each map $f :\equiv (f_0, f_p) : X \to_* Y$ of pointed 2-types, we have a 2-group morphism $\Omega(f) : \Omega(X) \to \Omega(Y)$ defined by induction on the path f_p.

As the next two examples show, not all 2-groups are defined directly as loop spaces, even though our main result, Theorem 6.11, implies all 2-groups are equivalent to them.

Example 4.8. For any bicategory C and $X : \mathrm{Ob}(C)$, the type of adjoint autoequivalences $\mathsf{AdjEquiv}(X, X)$ is a 2-group under composition of 1-cells [Aut-adj-2G]—the *automorphism* 2-*group on* X. If C is the bicategory of 1-types in a universe, then $\mathsf{AdjEquiv}(X, X)$ is the type of self-equivalences on X, and the function $\mathsf{univ} : (X \simeq X) \to (X = X)$ coming from univalence is a 2-group morphism [ua-2SGrpMap].

Example 4.9. The full subcategory of a monoidal groupoid C on the adjoint equivalences is a 2-group under C's tensor product, known as the *Picard* 2-*group* of C [13, Section 7].

We end this section with the variant of univalence for bicategories. As we'll see, this property interacts nicely with our concise definition of biequivalence (Definition 6.8).

Definition 4.10 ([1, Definition 3.1]). A bicategory C is *(globally) univalent* if the canonical function $(a = b) \to \mathsf{AdjEquiv}(a, b)$ is an equivalence for all $a, b : \mathrm{Ob}(C)$.

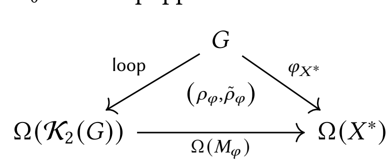
LEMMA 4.11 (ADJEQ-EXMPS). *Both* $\mathbf{2Type}_0^*$ *and* **c2Grp** *are univalent bicategories.*

PROOF. We factor through the SIP, which states that $a = b$ is equivalent, by sending refl_a to id_a, to *isomorphisms* $a \to b$, i.e., maps whose underlying functions are equivalences. □

5 Delooping a 2-group

The Eilenberg-MacLane space $K(H, 1)$ of an (axiomatic) group H [17, Section 3], also called the *classifying space* of H, is the 1-truncated HIT generated by $\mathsf{base} : K(H, 1)$, $\mathsf{loop} : H \to \mathsf{base} = \mathsf{base}$, and a term $\mathsf{loop\text{-}comp}$ witnessing that loop is a group map $H \to \Omega(K(H, 1))$. Let $\mathcal{U}$ be a universe and G be a 2-group in $\mathcal{U}$. Define the *classifying space* of G as the 2-truncated HIT $\mathcal{K}_2(G)$ generated by $\mathsf{base} : \mathcal{K}_2(G)$, $\mathsf{loop} : G \to \mathsf{base} = \mathsf{base}$, and two path constructors $\mathsf{loop\text{-}comp}$ and $\mathsf{loop\text{-}assoc}$ that make loop a 2-group map $G \to \Omega(\mathcal{K}_2(G))$ [Delooping]. An easy consequence of $\mathcal{K}_2$'s induction principle is that $\mathcal{K}_2(G)$ is connected [$\mathcal{K}_2$-is-conn].

The recursion principle for $\mathcal{K}_2$, which we derive from the induction principle, states that $\mathcal{K}_2(G)$ is initial in the category of pointed 2-types X^* equipped with a 2-group morphism $G \to \Omega(X^*)$. Explicitly, for every pointed 2-type $X^* :\equiv (X, x_0)$ together with a 2-group morphism $\varphi_{X^*} : G \to \Omega(X^*)$, we have a function $M_\varphi : \mathcal{K}_2(G) \to X$ that satisfies $M_\varphi(\mathsf{base}) \equiv x_0$ and is equipped with a natural isomorphism:



We call ρ_φ the *point β-rule* and $\tilde{\rho}_\varphi$ the *tensor β-rule* for M_φ.

A *delooping* of a group H is a pointed type B equipped with a group isomorphism $H \to \Omega(B)$. An essential property of $K(H,1)$ is that loop makes it a delooping of H. We want to prove that, likewise, loop makes $\mathcal{K}_2(G)$ a delooping of G, i.e., that loop is an equivalence of types. (In this case, we can easily show it's the unique delooping by $\mathcal{K}_2(G)$'s recursion principle.) To this end, we adapt the encode-decode proof used for $K(H,1)$ [17, Theorem 3.2] to the 2-dimensional case.

We define codes : $\mathcal{K}_2(G) \to \mathcal{U}^{\leq 1}$ by recursion on $\mathcal{K}_2(G)$ so that $\mathsf{pr}_1(\mathsf{codes}(\mathsf{base})) \equiv G$ [codes], where $\mathcal{U}^{\leq 1}$ denotes the 2-type of all 1-truncated types in $\mathcal{U}$. Since G is 1-truncated by definition, we take it as the basepoint of $\mathcal{U}^{\leq 1}$. To construct codes, we want a 2-group morphism $\zeta : G \to \Omega(\mathcal{U}^{\leq 1})$. Define $\zeta_{\mathsf{map}} : G \to (G = G)$ by mapping g to the type equivalence post-mult$(g) : G \xrightarrow{\simeq} G$ defined by post-mult$(g,x) :\equiv x \otimes g$ and then applying univ to it. Both post-mult and univ are 2-group morphisms (see [PostMultMap] and Example 4.8, respectively), and we give ζ_{map} the composite of their morphism structures. Now, consider the projection $\mathsf{codes}_0 :\equiv \mathsf{pr}_1 \circ \mathsf{codes} : \mathcal{K}_2(G) \to \mathcal{U}$. Define

$$\mathsf{encode} : (z : \mathcal{K}_2(G)) \to \mathsf{base} = z \to \mathsf{codes}_0(z)$$
$$\mathsf{encode}_z(p) :\equiv \mathsf{transp}^{\mathsf{codes}_0}(p, \mathsf{id}_G)$$

This yields $\mathsf{encode}_{\mathsf{base}} : \Omega(\mathcal{K}_2(G)) \to G$ [encode].

We want to show that loop : $G \to \Omega(\mathcal{K}_2(G))$ is an equivalence with inverse $\mathsf{encode}_{\mathsf{base}}$. As in [17], $\mathsf{encode}_{\mathsf{base}}$ is a left inverse of loop [encode-loop]. The main ingredient for the proof of this claim is the following chain of paths for all $x, y : G$, which we denote by transp-codes(x,y):

$$\mathsf{transp}^{\mathsf{codes}_0}(\mathsf{loop}(x), y)$$
$$= \quad \textit{via path induction on } \mathsf{loop}(x)$$
$$\mathsf{coe}(\mathsf{ap}_{\mathsf{pr}_1}(\mathsf{ap}_{\mathsf{codes}}(\mathsf{loop}(x))), y)$$
$$= \quad \textit{via } \mathsf{codes}\textit{'s point } \beta\textit{-rule}$$
$$\mathsf{coe}(\zeta_{\mathsf{map}}(x), y)$$
$$= \quad \textit{(typal) } \beta\textit{-rule for } \mathsf{coe}$$
$$y \otimes x$$

where coe : $(A = B) \to (A \simeq B)$, defined by path induction, is the inverse of univ and thus has a β-rule. This chain is also important for the next part of the proof, so we record the following coherence property satisfied by its final path.

Lemma 5.1 (coe-β-mu). *Figure 1 (below) commutes.*

Next, we show that $\mathsf{encode}_{\mathsf{base}}$ is a right inverse of loop. We want a homotopy ri : $\mathsf{loop} \circ \mathsf{encode}_{\mathsf{base}} \sim \mathsf{id}_{\Omega(\mathcal{K}_2(G))}$. We will define decode : $(z : \mathcal{K}_2(G)) \to \mathsf{codes}_0(z) \to \mathsf{base} = z$ by induction on $\mathcal{K}_2(G)$ so that decode(base) $\equiv$ loop. By path induction, we then see that $\mathsf{decode}_z(\mathsf{encode}_z(p)) = p$ for all $z : \mathcal{K}_2(G)$ and $p : \mathsf{base} = z$ because every 2-group morphism preserves the unit. This gives us ri by setting z to base.

We now describe the construction of decode [Decode-def], which is much more complex than the 1-dimensional case. Here, the target of the induction is the function type $\mathsf{codes}_0(z) \to \mathsf{base} = z$ for all $z : \mathcal{K}_2(G)$. In such a situation, we have the following form of the induction principle.

Lemma 5.2 (PPOverFun). *Let B_1 be a type family over $\mathcal{K}_2(G)$ and B_2 a family of 1-types over $\mathcal{K}_2(G)$. Suppose we have a function $\psi_{\mathsf{base}} : B_1(\mathsf{base}) \to B_2(\mathsf{base})$ together with*

- *an identity $\psi_{\mathsf{loop}}(x,b) : \psi_{\mathsf{base}}(\mathsf{transp}^{B_1}(\mathsf{loop}(x), b)) = \mathsf{transp}^{B_2}(\mathsf{loop}(x), \psi_{\mathsf{base}}(b))$ for each $x : G$ and $b : B_1(\mathsf{base})$*
- *for all $x, y : G$ and $b : B_1(\mathsf{base})$, the commuting diagram displayed by Fig. 2.*

Then we have a function $\psi : (x : \mathcal{K}_2(G)) \to B_1(x) \to B_2(x)$ such that $\psi(\mathsf{base}) \equiv \psi_{\mathsf{base}}$.

Lemma 5.2 avoids the input data for loop-assoc because the induction's target is a 1-type. By instantiating B_1 with $\mathsf{codes}_0(z)$ and B_2 with base $= z$, Lemma 5.2 gives a sufficient condition for building decode, namely the data ψ_{base}, ψ_{loop}, and $\psi_{\mathsf{loop\text{-}comp}}$. Of course, we define ψ_{base} as loop. For $x, y : G$, we define $\psi_{\mathsf{loop}}(x,y)$ as the chain

$$\mathsf{loop}(\mathsf{transp}^{\mathsf{codes}_0}(\mathsf{loop}(x), y))$$
$$= \quad \mathsf{ap}_{\mathsf{loop}}(\mathsf{transp\text{-}codes}(x,y))$$
$$\mathsf{loop}(y \otimes x)$$
$$= \quad \mathsf{loop\text{-}comp}(y,x)$$
$$\mathsf{loop}(y) \cdot \mathsf{loop}(x)$$
$$= \quad \textit{behavior of } \mathsf{transp} \textit{ in path families}$$
$$\mathsf{transp}^{z \mapsto \mathsf{base}=z}(\mathsf{loop}(x), \mathsf{loop}(y))$$

Finally, we construct $\psi_{\mathsf{loop\text{-}comp}}$ (which doesn't show up in [17]'s 1-dimensional setting). Let $x, y, z : G$. We want to prove that the outer diagram of Fig. 3 commutes. The subdiagrams of Fig. 3 commute as follows: S_1 and S_2 by homotopy naturality at transp-codes; S_3 and S_5 by homotopy naturality at loop-comp; S_4 by path induction; and S_6 by loop-assoc.

It remains to build a path transp-codes-coh filling S_7, at the top of Fig. 3. By unfolding S_7, we see that this path fills Fig. 4: the image under $\mathsf{ap}_{\mathsf{loop}}$ of a diagram D of paths in G. Thus, it suffices to fill D. The bottom left corner of D fits into the commuting square Fig. 5. After using this square to rewrite D, we rewrite the three paths making up the top

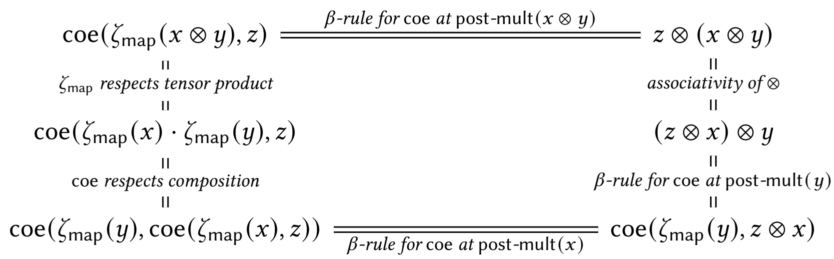


Figure 1: coherence property of final path in transp-codes **for all** $x, y, z : G$

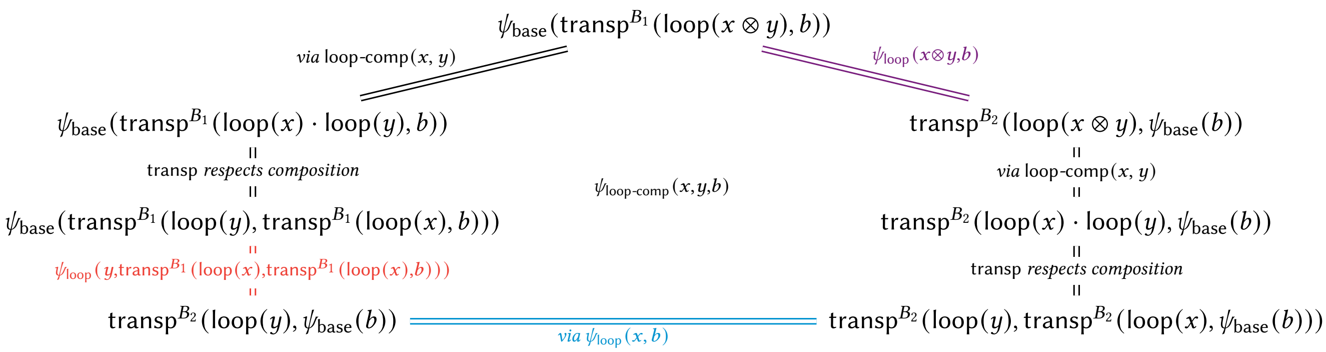


Figure 2: coherence condition, labeled $\psi_{\text{loop-comp}}$**, for** loop-comp **at** x, y, **and** b **(the colors are for readability of Fig. 3)**

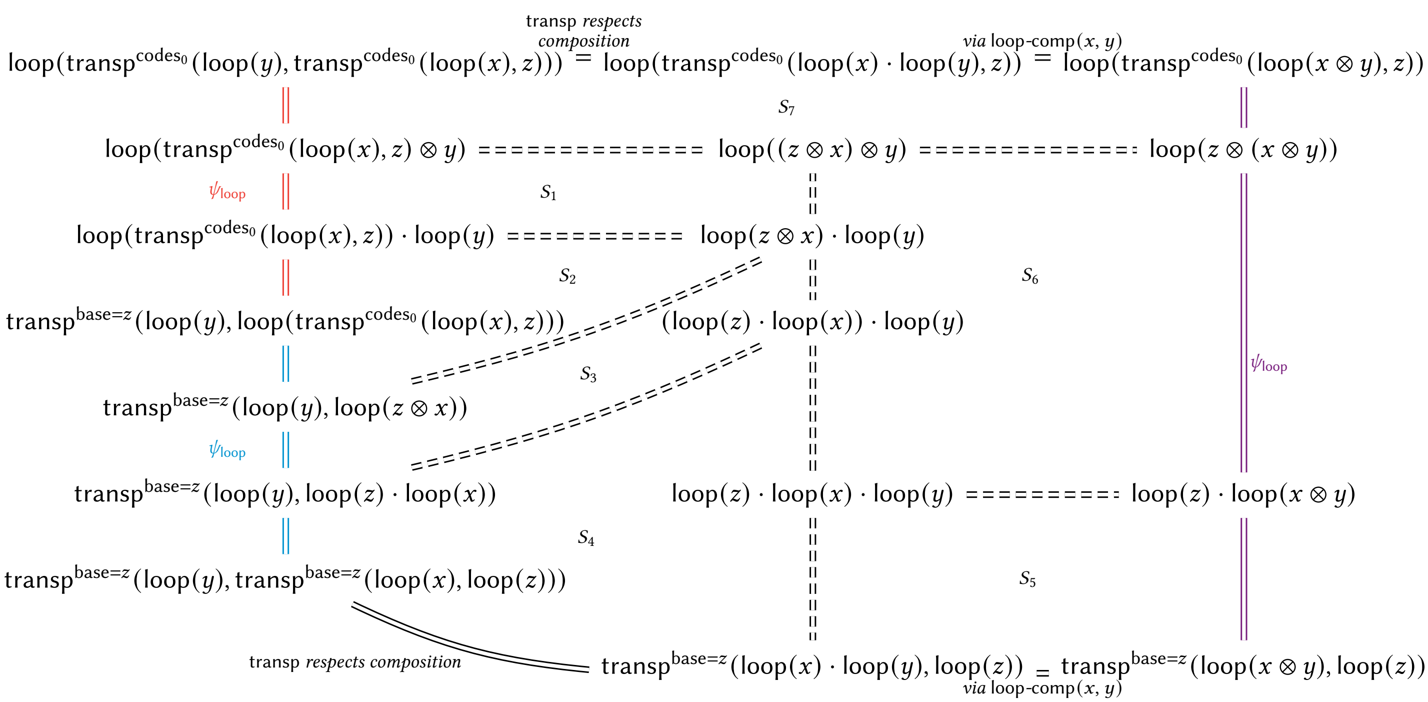


Figure 3: construction of $\psi_{\text{loop-comp}}(x, y, z)$**, where the colors match those of Fig. 2 for readability**

$$\begin{array}{ccc}
\mathsf{loop}(\mathsf{transp}^{\mathsf{codes}_0}(\mathsf{loop}(x)\cdot\mathsf{loop}(y),z)) & \overset{\textit{via}\ \mathsf{loop\text{-}comp}(x,\,y)}{=\!=\!=\!=\!=} & \mathsf{loop}(\mathsf{transp}^{\mathsf{codes}_0}(\mathsf{loop}(x\otimes y),z)) \\
\| & & \| \\
\mathsf{transp}\ \textit{respects composition} & & \mathsf{ap}_{\mathsf{loop}}(\mathsf{transp\text{-}codes}(x\otimes y,z)) \\
\| & & \| \\
\mathsf{loop}(\mathsf{transp}^{\mathsf{codes}_0}(\mathsf{loop}(y),\mathsf{transp}^{\mathsf{codes}_0}(\mathsf{loop}(x),z))) & & \mathsf{loop}(z\otimes(x\otimes y)) \\
\| & & \| \\
\mathsf{ap}_{\mathsf{loop}}(\mathsf{transp\text{-}codes}(y,\mathsf{transp}^{\mathsf{codes}_0}(\mathsf{loop}(x),z))) & & \textit{associativity of}\otimes \\
\| & & \| \\
\mathsf{loop}(\mathsf{transp}^{\mathsf{codes}_0}(\mathsf{loop}(x),z)\otimes y) & \underset{\textit{via}\ \mathsf{transp\text{-}codes}(x,\,z)}{=\!=\!=\!=\!=} & \mathsf{loop}((z\otimes x)\otimes y)
\end{array}$$

Figure 4: the diagram filled by transp-codes-coh

$$\begin{array}{ccc}
\mathsf{transp}^{\mathsf{codes}_0}(\mathsf{loop}(y),\mathsf{transp}^{\mathsf{codes}_0}(\mathsf{loop}(x),z)) & \overset{\textit{via}\ \mathsf{transp\text{-}codes}(x,\,z)}{=\!=\!=\!=\!=} & \mathsf{transp}^{\mathsf{codes}_0}(\mathsf{loop}(y),z\otimes x) \\
\| & & \| \\
\mathsf{transp\text{-}codes}(y,\mathsf{transp}^{\mathsf{codes}_0}(\mathsf{loop}(x),z)) & \mathsf{hnat}(\mathsf{transp\text{-}codes}(x,z)) & \mathsf{transp\text{-}codes}(y,z\otimes x) \\
\| & & \| \\
\mathsf{transp}^{\mathsf{codes}_0}(\mathsf{loop}(x),z)\otimes y & \underset{\textit{via}\ \mathsf{transp\text{-}codes}(x,\,z)}{=\!=\!=\!=\!=} & (z\otimes x)\otimes y
\end{array}$$

Figure 5: rewriting $\mathsf{transp}^{\mathsf{codes}_0}(\mathsf{loop}(x),z)\otimes y$

right corner of D, namely transp-codes$(x\otimes y,z)$:

$$\begin{array}{c}
\mathsf{transp}^{\mathsf{codes}_0}(\mathsf{loop}(x\otimes y),z) \\
\| \\
\textit{via path induction on}\ \mathsf{loop}(x\otimes y) \\
\| \\
\mathsf{coe}(\mathsf{ap}_{\mathsf{pr}_1}(\mathsf{ap}_{\mathsf{codes}}(\mathsf{loop}(x\otimes y))),z) \\
\| \\
\textit{via}\ \mathsf{codes}\textit{'s point}\ \beta\textit{-rule} \\
\| \\
\mathsf{coe}(\zeta_{\mathsf{map}}(x\otimes y),z) \\
\| \\
\beta\textit{-rule for}\ \mathsf{coe} \\
\| \\
z\otimes(x\otimes y)
\end{array}$$

Call these p_1, p_2, and p_2, respectively. Rewrite p_1 with homotopy naturality as in the commuting square Fig. 6. Rewrite p_2 with codes's tensor β-rule [codes-β-mu] to get the commuting diagram Fig. 7. Lastly, rewrite p_3 with Lemma 5.1.

Now, we fill D by cancelling sets of like terms, such as codes's point β-rules. This completes decode. Hence loop is an equivalence [loop-equiv].

Remark 5.3. Our delooping proof extends [6, Section 4.3], which shows the result when G is a 1-group. The difference between the two proofs is that when G is a 1-group,

- the target of codes is the 1-type $\mathcal{U}^{\leq 0}$ instead of $\mathcal{U}^{\leq 1}$
- defining transp-codes-coh is trivial as G is a set.

Our formalization is entirely separate from that of [6].

6 The loop space as a biequivalence

In this section, we make $\mathcal{K}_2$ into a *pseudofunctor*, which extends the notion of functor to bicategories. Then we use Section 5 to show that, with $\mathcal{K}_2$, the loop space pseudofunctor forms a biadjoint biequivalence between $\mathbf{2Type}_{\mathbf{0}}^{*}$ and $\mathbf{c2Grp}$.

Definition 6.1 (PsftorStr). A *pseudofunctor* $\mathcal{C}\to\mathcal{D}$ between bicategories is a function $F_0:\mathrm{Ob}(\mathcal{C})\to\mathrm{Ob}(\mathcal{D})$ with

- a function $F_1:\mathrm{hom}_{\mathcal{C}}(a,b)\to\mathrm{hom}_{\mathcal{D}}(F_0(a),F_0(b))$ for all $a,b:\mathrm{Ob}(\mathcal{C})$, called the *action on morphisms*
- a 2-cell $F\text{-}\mathrm{id}_a:F_1(\mathrm{id}_a)=\mathrm{id}_{F_0(a)}$ for each $a:\mathrm{Ob}(\mathcal{C})$
- a 2-cell $F_\circ(f,g):F_1(g\circ f)=F_1(g)\circ F_1(f)$ for all composable morphisms f and g
- coherence identities making $F_\circ$ commute with the right unitors, the left unitors, and the associators.

Example 6.2. The loop space Ω forms a pseudofunctor $\mathbf{2Type}_{\mathbf{0}}^{*}\to\mathbf{c2Grp}$, whose object function and action on morphisms are defined as in Example 4.7. By the SIP for pointed homotopies, we can put the action on 2-cells in an extensional form 2c-act$_\Omega$ [LoopFunctor-ap] that takes pointed homotopies to natural isomorphisms.

Lemma 6.3 (Ω-fmap-ap-hnat). *Let* $f:=(f_0,f_p),g:=(g_0,g_p):(X,x_0)\to Y$ *be maps in* $\mathbf{2Type}_{\mathbf{0}}^{*}$. *Let* $H:=(H_0,H_p)$ *be a pointed homotopy* $f\sim_* g$. *The underlying homotopy* θ_H *of* 2c-act$_\Omega(H)$ *fits into a commuting pentagon for all* $p:\Omega(X)$:

$$\begin{array}{ccc}
\mathsf{fun}(\Omega(f))(p) & \overset{\theta_H(p)}{=\!=\!=\!=\!=} & \mathsf{fun}(\Omega(g))(p) \\
\| & & \| \\
\textit{typal}\ \beta\textit{-rule for}\ \Omega(f) & & \textit{typal}\ \beta\textit{-rule for}\ \Omega(g) \\
\| & & \| \\
f_p^{-1}\cdot\mathsf{ap}_f(p)\cdot f_p & & g_p^{-1}\cdot\mathsf{ap}_g(p)\cdot g_p \\
\textit{via}\ \mathsf{hnat}(p)\ \searrow\!\!\searrow & & \swarrow\!\!\swarrow\ \textit{via}\ H_p \\
& f_p^{-1}\cdot\left(H_0(x_0)\cdot\mathsf{ap}_g(p)\cdot H_0(x_0)^{-1}\right)\cdot f_p &
\end{array}$$

$$\begin{array}{ccc}
\mathsf{transp}^{\mathsf{codes}_0}(\mathsf{loop}(x \otimes y), z) & \overset{p_1}{=\!=\!=\!=} & \mathsf{coe}(\mathsf{ap}_{\mathsf{pr}_1}(\mathsf{ap}_{\mathsf{codes}}(\mathsf{loop}(x \otimes y))), z) \\
\| & \mathsf{hnat}(\mathsf{loop\text{-}comp}(x,y)) & \| \\
\mathsf{transp}^{\mathsf{codes}_0}(\mathsf{loop}(x) \cdot \mathsf{loop}(y), z) & =\!=\!= & \mathsf{coe}(\mathsf{ap}_{\mathsf{pr}_1}(\mathsf{ap}_{\mathsf{codes}}(\mathsf{loop}(x) \cdot \mathsf{loop}(y))), z)
\end{array}$$

Figure 6: rewriting p_1, the path defined via path induction on $\mathsf{loop}(x \otimes y)$

$$\begin{array}{ccc}
\mathsf{ap}_{\mathsf{pr}_1}(\mathsf{ap}_{\mathsf{codes}}(\mathsf{loop}(x \otimes y))) & \overset{p_2}{=\!=\!=\!=\!=\!=} & \zeta_{\mathsf{map}}(x \otimes y) \\
\| & & \| \;\; \zeta_{\mathsf{map}} \textit{ respects tensor product} \\
\| \;\; \textit{via } \mathsf{loop\text{-}comp}(x, y) & & \zeta_{\mathsf{map}}(x) \cdot \zeta_{\mathsf{map}}(y) \\
\| & & \| \;\; \textit{via } \mathsf{codes}\textit{'s point } \beta\textit{-rule at } y \\
\mathsf{ap}_{\mathsf{pr}_1}(\mathsf{ap}_{\mathsf{codes}}(\mathsf{loop}(x))) \cdot \mathsf{ap}_{\mathsf{pr}_1}(\mathsf{ap}_{\mathsf{codes}}(\mathsf{loop}(y))) & \underset{\textit{via } \mathsf{codes}\textit{'s point } \beta\textit{-rule at } x}{=\!=\!=\!=} & \zeta_{\mathsf{map}}(x) \cdot \mathsf{ap}_{\mathsf{pr}_1}(\mathsf{ap}_{\mathsf{codes}}(\mathsf{loop}(y)))
\end{array}$$

Figure 7: rewriting p_2, the path defined via codes**'s point β-rule**

Turning to $\mathcal{K}_2$, the next two lemmas follow from its induction principle. The first gives a way to build a homotopy between two functions defined by $\mathcal{K}_2$-recursion, and the second a way to prove that two such homotopies are equal. The first lemma will be useful for defining 2-cells in $\mathbf{2Type}_{\mathbf{0}}^*$, and the second for proving coherence conditions on them.

Lemma 6.4 (K-hom-ind). *Let G be a 2-group and X be a 2-type. Let $f, g : \mathcal{K}_2(G) \to X$. Given terms*

$\mathsf{base}^\sim : f(\mathsf{base}) = g(\mathsf{base})$

$\mathsf{loop}^\sim : (x : G) \to \mathsf{ap}_f(\mathsf{loop}(x)) \cdot \mathsf{base}^\sim = \mathsf{base}^\sim \cdot \mathsf{ap}_g(\mathsf{loop}(x))$

$\mathsf{loop\text{-}comp}^\sim : \mathsf{loop}^\sim$ *commutes with G's tensor product*

we have a homotopy $H : f \sim g$ satisfying $H(\mathsf{base}) \equiv \mathsf{base}^\sim$ and the following typal β-rule for all $x : G$:

$$\begin{array}{ccc}
f(\mathsf{base}) & =\!=\!= & g(\mathsf{base}) \\
\| & \mathsf{hnat}(\mathsf{loop}(x)) & \| \\
f(\mathsf{base}) & =\!=\!= & g(\mathsf{base})
\end{array}
\quad = \quad
\begin{array}{ccc}
f(\mathsf{base}) & =\!=\!= & g(\mathsf{base}) \\
\| & \mathsf{loop}^\sim(x) & \| \\
f(\mathsf{base}) & =\!=\!= & g(\mathsf{base})
\end{array}$$

Lemma 6.5 (K-hom2-ind). *Let G, X, f, and g be as in Lemma 6.4. Let $H_1, H_2 : f \sim g$. Suppose we have an identity $\mathsf{base}^{\sim\sim} : H_1(\mathsf{base}) = H_2(\mathsf{base})$ and a 3-dimensional path $\mathsf{loop}^{\sim\sim}$ whose type is displayed by Fig. 8. Then we have a homotopy $R : H_1 \sim H_2$ such that $R(\mathsf{base}) \equiv \mathsf{base}^{\sim\sim}$.*

Example 6.6. We equip $\mathcal{K}_2 : \mathrm{Ob}(\mathbf{c2Grp}) \to \mathrm{Ob}(\mathbf{2Type}_{\mathbf{0}}^*)$ with the structure of a pseudofunctor. Its action on morphisms [$\mathcal{K}_2$-map] sends $f : G_1 \to G_2$ to the pointed map defined by $\mathcal{K}_2$-recursion on $\mathsf{loop} \circ f : G_1 \to \Omega(\mathcal{K}_2(G_2))$. This action preserves the identity morphism [KFunctor-idf] and composition [KFunctor-comp], with both preservation proofs defined via Lemma 6.4. We use Lemma 6.5 to prove coherence with unitors [KFunctor-conv-unit] and coherence with the associator [KFunctor-conv-assoc].

As for Ω, we can put $\mathcal{K}_2$'s action on 2-cells in an extensional form $\mathsf{2c\text{-}act}_{\mathcal{K}_2}$—defined via Lemma 6.4—that takes natural isomorphisms to pointed homotopies [ap$\mathcal{K}_2$].

Definition 6.7 (Pstransf). For pseudofunctors $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{C}$, a *pseudotransformation* $F \Rightarrow G$ consists of

- for each $a : \mathrm{Ob}(\mathcal{C})$, a 1-cell $\xi_0(a) : F_0(a) \to G_0(a)$, called a *component* of the pseudotransformation
- for all $f : \hom_{\mathcal{C}}(a, b)$, a 2-cell $\xi_1(f)$ filling the square

$$\begin{array}{ccc}
F_0(a) & \xrightarrow{F_1(f)} & F_0(b) \\
{\scriptstyle \xi_0(a)}\downarrow & & \downarrow{\scriptstyle \xi_0(b)} \\
G_0(a) & \xrightarrow[G_1(f)]{} & G_0(b)
\end{array}$$

- coherence identities witnessing that ξ_1 commutes with the unitors and with the associators.

A pseudtransformation is a *pseudoequivalence* if all its components are adjoint equivalences.

Note that a pseudofunctor automatically commutes with 2-cells by homotopy naturality.

Definition 6.8 (Biequiv). A *(biadjoint) biequivalence between $\mathcal{C}$ and $\mathcal{D}$* is a pseudofunctor $F : \mathcal{C} \to \mathcal{D}$ along with

- a pseudofunctor $G : \mathcal{D} \to \mathcal{C}$
- a pseudoequivalence $\varepsilon : G \circ F \Rightarrow \mathrm{id}_{\mathcal{C}}$
- a pseudoequivalence $\eta : \mathrm{id}_{\mathcal{D}} \Rightarrow F \circ G$

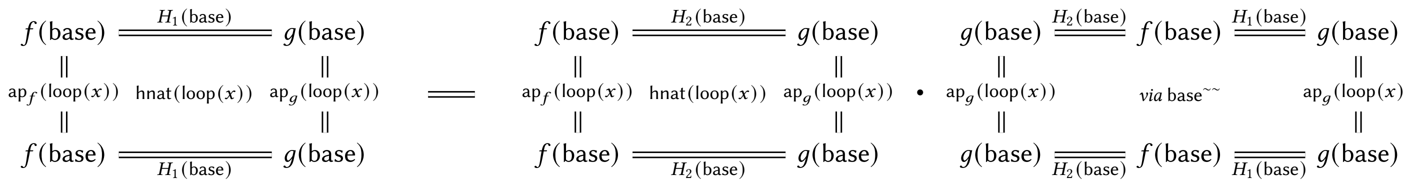
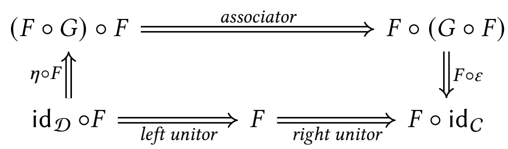


Figure 8: type of $\mathsf{loop}^{\sim\sim}$**, where the operation • denotes horizontal composition of squares**

- a path, called the *triangulator*, filling the diagram



Note that the triangulator is equivalent to an *invertible modification* [1, Definition 2.14], that is, a family of paths $(\mathsf{id}_{\mathcal{D}} \circ F)_0(a) = (F \circ \mathsf{id}_{C})_0(a)$ that is natural in $a : \mathrm{Ob}(C)$.

One might wonder why Definition 6.8 omits the other triangle identity and the swallowtail identities, which are included in a coherent biadjunction. With fewer fields, it makes building biequivalences easier. Also, it is equivalent, in a coherent sense, to the one specifying all the biadjunction data. A classical proof of Gurski's contains such an equivalence [11, Theorem 3.2], and it seems Gurski's argument could be ported to HoTT. Within HoTT, when C and $\mathcal{D}$ are univalent (as in the scenario we care about), our short definition is fully coherent in the following sense.

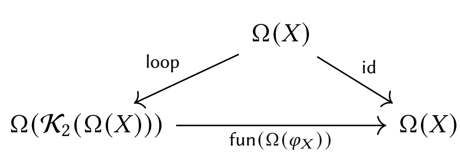
Lemma 6.9 (biadjequiv-is-prop). *Let* $F : C \to \mathcal{D}$ *be a pseudofunctor between univalent bicategories. The type* (F is a biequivalence) *is a proposition.*

Proof. By univalence, both η and ε become paths. Thus, biequivalence data on F behaves like half-adjoint equivalence data on a function, and the latter is a proposition. □

With Lemma 6.9, we will see that our notion of biequivalence is the same as isomorphism of bicategories (Lemma 7.3).

Returning to our desired biequivalence, the next result gives one of the families of adjoint equivalences that Definition 6.8 requires. (Section 5 gives the other.)

Note 6.10. Let X be a pointed connected 2-type. Define the pointed map $\varphi_X : \mathcal{K}_2(\Omega(X)) \to_* X$ by $\mathcal{K}_2$-recursion on the identity 2-group morphism $\Omega(X) \to \Omega(X)$. By φ_X's point β-rule, the following triangle commutes [LoopK-hom]:



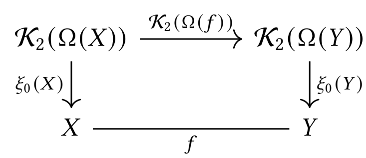
By Section 5, loop is an equivalence, so that $\mathsf{fun}(\Omega(\varphi_X))$ is one as well. Since both X and $\mathcal{K}_2(\Omega(X))$ are connected, it follows that φ_X is an equivalence [Loop-conn-equiv].

Theorem 6.11 (Biadj-biequiv-main). *The pseudofunctors* Ω *and* $\mathcal{K}_2$ *form a biequivalence between* $\mathbf{2Type}_0^*$ *and* $\mathbf{c2Grp}$.

Proof. We outline the four major steps of the proof.

Step 1: *Construct* $\varepsilon : \mathcal{K}_2 \circ \Omega \Rightarrow \mathsf{id}_{\mathbf{2Type}_0^*}$.

For every pointed connected 2-type X, define the map $\xi_0(X) : \mathcal{K}_2(\Omega(X)) \to_* X$ as φ_X (see Note 6.10). Let $f : X \to Y$ be a map in $\mathbf{2Type}_0^*$. We want a path $\xi_1(f)$ making the following square commute:



By the SIP for pointed maps, it suffices to find a pointed homotopy $(H_1(f), H_2(f)) : f \circ \xi_0(X) \sim_* \xi_0(Y) \circ \mathcal{K}_2(\Omega(f))$. We define $H_1(f) : \mathsf{fun}(f) \circ \mathsf{fun}(\xi_0(X)) \sim \mathsf{fun}(\xi_0(Y)) \circ \mathsf{fun}(\mathcal{K}_2(\Omega(f)))$ by applying Lemma 6.4 to $\mathsf{base}^{\sim} :\equiv \mathsf{refl}$, $\mathsf{loop}^{\sim} :\equiv H_1(f)\text{-}\,\mathsf{loop}$, and $\mathsf{loop\text{-}comp}^{\sim} :\equiv H_1(f)\text{-}\mathsf{loop\text{-}comp}$. Here, $H_1(f)\text{-}\,\mathsf{loop}(p)$ is defined as the chain

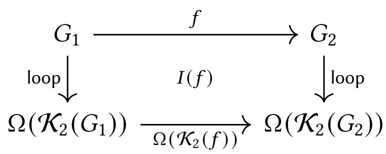
$$\begin{aligned}
&\mathsf{ap}_{\mathsf{fun}(f \circ \xi_0(X))}(\mathsf{loop}(p)) \\
&= \quad \textit{via } \xi_0(X)\textit{'s point } \beta\textit{-rule} \\
&\mathsf{ap}_{\mathsf{fun}(f)}(p) \\
&= \quad \textit{via } \xi_0(Y)\textit{'s point } \beta\textit{-rule} \\
&\mathsf{ap}_{\mathsf{fun}(\xi_0(Y))}(\mathsf{loop}(\mathsf{ap}_{\mathsf{fun}(f)}(p))) \\
&= \quad \textit{via } \mathcal{K}_2(\Omega(f))\textit{'s point } \beta\textit{-rule} \\
&\mathsf{ap}_{\mathsf{fun}(\xi_0(Y) \circ \mathcal{K}_2(\Omega(f)))}(\mathsf{loop}(p))
\end{aligned}$$

for each $p : x_0 = x_0$ where x_0 denotes the basepoint of X. The term $H_1(f)$-loop-comp is a routine yet long computation, and we refer the reader to its mechanization [SqKLoop-coher]. Our definition of $H_1(f)$ makes it trivial to define $H_2(f)$, so the definition of $\xi_1(f)$ is complete [SqKLoop].

In Section B, we prove coherence of ξ_1 via Lemma 6.5.

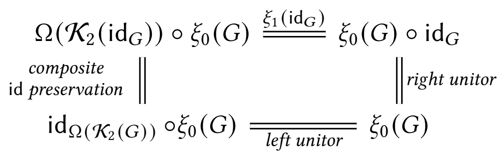
Step 2: *Construct* $\eta : \mathsf{id}_{\mathbf{c2Grp}} \Rightarrow \Omega \circ \mathcal{K}_2$.

For each 2-group G, define the 2-group morphism $\xi_0(G) : G \to \Omega(\mathcal{K}_2(G))$ as loop. Let $f : G_1 \to G_2$ be a 2-group map. To define the path $\xi_1(f)$, we want a natural isomorphism:

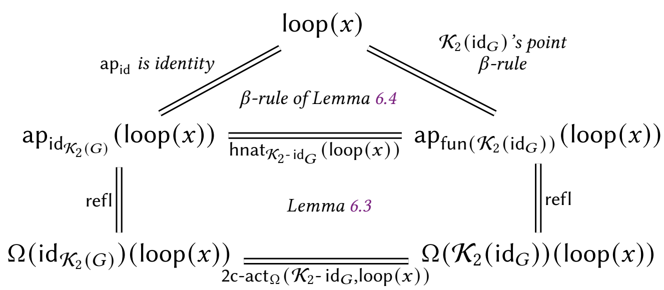


We define the two components of $I(f)$ from $\mathcal{K}_2(f)$'s point and tensor β-rules, respectively [SqLoopK].

We prove that ξ_1 satisfies the relevant coherence conditions. In the case of unitors, we want to prove that



commutes for all 2-groups G. By the SIP for natural isos, this square amounts to a homotopy H_G between the underlying homotopies of the associated natural isomorphisms. For each $x : G$, we define $H_G(x)$ as the following commuting outer diagram, where 2c-act$_\Omega$ is as in Example 6.2:



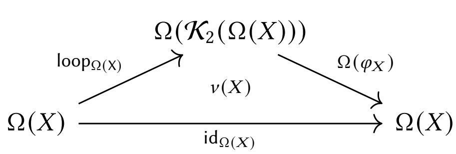
This completes the coherence identity with the unitors [LoopK-PT-unit]. The coherence identity with the associator is similar but more complicated [LoopK-PT-assoc].

Step 3: *Prove that ε and η are levelwise adjoint equivalences.*

By Note 6.10, ε is a levelwise adjoint equivalence. By Section 5, η is a levelwise adjoint equivalence.

Step 4: *Construct the triangulator.*

We construct an invertible modification between $\eta \circ \Omega$ and $\Omega \circ \varepsilon$ (hiding associativity and unit terms for readability) [Loop-zig-zag]. We first need a natural isomorphism



for each pointed connected 2-type X, where φ_X is as in Note 6.10. We define $\nu(X)$ directly from φ_X's β-rules [Loop-zz$_0$-iso]. We also need to prove ν is natural in X, i.e., that Fig. 9 commutes for every map $f : X \to Y$ in $\mathbf{2Type}_0^*$. By the SIP for natural isos, we just need an identity between the relevant underlying homotopies. We get one by applying Lemma 6.3 and then the β-rule of Lemma 6.4 to the left arrow of Fig. 9 [Loop-zz$_1$-$\sim$]. □

Remark 6.12. In general, one can adjust the unit or counit of an incoherent biequivalence to get the triangulator [11, Theorem 3.2]. For us, however, this process would conceivably make the new pseudotransformation harder to work with. For example, the components ξ_0 from Steps 1 and 2 have simple forms—unlike their inverses, which would be included in the adjusted pseudotransformation.

7 The loop space as an isomorphism

In this section, we prove that the pseudofunctor $\Omega : \mathbf{2Type}_0^* \to \mathbf{c2Grp}$ is an *isomorphism*, i.e., an equivalence on objects and fully faithful. We do so by proving that a pseudofunctor between univalent bicategories is a biequivalence precisely when it is an isomorphism. The SIP tells us that isomorphism captures the notion of identity, so we can view this proof as justification for our short definition of biequivalence.

Definition 7.1. Let C and $\mathcal{D}$ be bicategories and let $F : C \to \mathcal{D}$ be a pseudofunctor. We say that F is an *isomorphism* if $F_0 : \mathrm{Ob}(C) \to \mathrm{Ob}(\mathcal{D})$ is an equivalence and $F_1 : \mathrm{hom}_C(a, b) \to \mathrm{hom}_{\mathcal{D}}(F_0(a), F_0(b))$ is an equivalence for all $a, b : \mathrm{Ob}(C)$. We denote the type of isomorphisms by $\cong$.

Lemma 7.2 (iso-bc-==-$\simeq$). *For all bicategories C and $\mathcal{D}$, the canonical function $(C = \mathcal{D}) \to (C \cong \mathcal{D})$ is an equivalence.*

Proof. By the SIP along with the univalence axiom. □

Lemma 7.3 (bae-iso-$\simeq$). *A pseudofunctor of univalent bicategories is a biequivalence if and only if it is an isomorphism.*

Proof. Let $F : C \to \mathcal{D}$ be a biequivalence of univalent bicategories. Since C and $\mathcal{D}$ are univalent, it's easy to see that F is an equivalence on objects. To see that F is fully faithful, we reuse the standard proof that adjoint equivalences of 1-categories are fully faithful.

Conversely, by Lemma 7.2, it suffices to observe that the identity pseudofunctor is a biequivalence. □

Theorem 7.4. *The pseudofunctor Ω is an isomorphism.*

Proof. By Lemma 7.3 and Theorem 6.11. □

Corollary 7.5 (Equality-main). *The pseudofunctor Ω induces a path $\mathbf{2Type}_0^* = \mathbf{c2Grp}$.*

By the same reasoning, $\mathcal{K}_2$ induces a path $\mathbf{c2Grp} = \mathbf{2Type}_0^*$.

Remark 7.6. The proof that a biequivalence is fully faithful works on the level of *wild categories* [9, Appendix B]. Thus, to derive the isomorphism $\mathbf{2Type}_0^* \cong \mathbf{c2Grp}$, we use only that Ω is a pseudofunctor and that it forms a 1-coherent adjoint

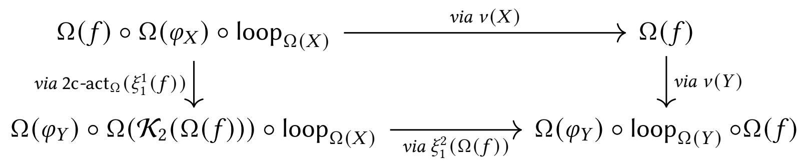


Figure 9: naturality condition for ν, where ξ_1^1 and ξ_1^2 are the 2-cells—as natural isos—from Steps 1 and 2, respectively

equivalence with $\mathcal{K}_2$. This means we can avoid most of the hardest computations making up Theorem 6.11! But this approach puts the cart before the horse: We don't know a priori whether the 1-coherent equivalence is part of a biequivalence, so the biequivalence we would get from the isomorphism—by the general method, adapted to the univalent setting, of turning a weak equivalence into a biequivalence [16, Theorem 7.4.1]—would have a less tractable form than Theorem 6.11.

8 Delooping types

We move to our second contribution. In Section 5, we considered deloopings of types carrying the structure of a coherent 2-group—a generalization of delooping 1-groups. To classify pointed connected 2-types by Sính triples, we need to consider a wider class of deloopings. We review some results—on the existence and uniqueness of deloopings—to this effect.

Definition 8.1. Let X be a pointed type and let $\mathcal{U}$ be a universe. The type of X-*torsors* is

$$TX :\equiv (Y : \mathcal{U}) \times \|Y\| \times ((y : Y) \to X \simeq_* (Y, y))$$

where $\simeq_*$ denotes the type of pointed equivalences.

Theorem 8.2 (torsors.Delooping). *Suppose the type $T_*X :\equiv (\tau : TX) \times \mathsf{pr}_1(\tau)$ is contractible with center $(\tau_c(X), b_X)$.*

(1) For all $\tau : TX$, $(\tau = \tau_c(X)) \simeq \mathsf{pr}_1(\tau)$.
(2) The pointed type $(TX, \tau_c(X))$ is a delooping of X, i.e., it admits a pointed equivalence $\Omega(TX) \simeq_* X$.
(3) Let Z be a pointed connected type in $\mathcal{U}$. If Z is a delooping of X, then $Z \simeq_* TX$.

Proof. By the proof of [32, Theorem 2]. ◻

Example 8.3. If X is m-connected and $2m$-truncated with $m \geq -1$, then T_*X is contractible [32, Corollary 7]. For each abelian group A and integer $n \geq 1$, recall that the Eilenberg-MacLane space $K(A, n)$ [17, Section 5] is $(n-1)$-connected and n-truncated and that $\Omega(K(A, n+1)) \simeq_* K(A, n)$. Thus, if $n \geq 2$, we have a pointed equivalence $K(A, n+1) \simeq_* T(K(A, n))$ [EM-Torsors-≃].

Theorem 8.4 ([7, Theorem 5.1]). *If $n \geq 2$, $K(-, n) : \mathbf{Ab} \to \mathcal{U}_*^{\geq n-1, \leq n}$ is an equivalence of types with inverse π_n, where $\mathcal{U}_*^{\geq n-1, \leq n}$ denotes the type of $(n-1)$-connected, n-truncated pointed types in $\mathcal{U}$ and π_n denotes the n-th homotopy group.*

Corollary 8.5 (EM-Ω-==-ext). *Let A be an abelian group and X a pointed type. For $n \geq 2$, $K(A, n) \simeq_* X$ is equivalent to $(X$ is $(n-1)$-connected$) \times (X$ is n-truncated$) \times (\pi_n(X) = A)$.*

Proof. If X is $(n-1)$-connected and n-truncated, then $(K(A, n) = X) \simeq (\pi_n(X) = A)$ by Theorem 8.4. ◻

9 Classification by Sính triples

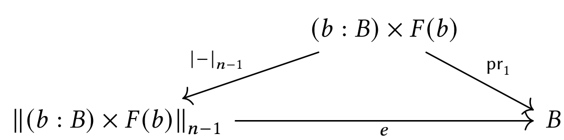
Let n be a positive integer. Recall that, according to the uniform definition of higher groups in HoTT [7], the internal n-groups are the pointed connected n-types. So, the type of (internal) n-groups is exactly $\mathcal{U}_*^{\geq 0, \leq n}$, and $\mathcal{U}_*^{\geq 0, \leq 2}$ is exactly the type of objects of the bicategory $\mathbf{2Type}_0^*$. In this section, using Section 8, we classify $(n+1)$-groups in terms of group cohomology (Theorem 9.2). To begin, we decompose any n-group into three pieces of data as follows.

Lemma 9.1 (N-Grps-≃). *We have an equivalence*

$$\mathcal{U}_*^{\geq 0, \leq n} \simeq \left(B : \mathcal{U}_*^{\geq 0, \leq n-1}\right) \times \left(F : B \to \mathcal{U}^{\geq n-1, \leq n}\right) \times F(\mathsf{pt}_B)$$

Proof. In one direction, send an n-group $G : \mathcal{U}_*^{\geq 0, \leq n}$ to the triple $\delta(G) :\equiv (B, F, p)$ defined as follows. Let $B :\equiv \|G\|_{n-1}$ with basepoint $\left|\mathsf{pt}_G\right|_{n-1}$. This is connected because truncations preserve connectedness. Let $F(b) :\equiv \mathsf{fib}_{|-|_{n-1}}(b)$, the fiber of $|-|_{n-1}$ over b. This is $(n-1)$-connected by [29, Corollary 7.5.8]. It is n-truncated because Σ preserves n-types. Let $p :\equiv (\mathsf{pt}_G, \mathsf{refl})$. In the other direction, send (B, F, p) to the pointed n-type $\alpha(B, F, p) :\equiv \left((b : B) \times F(b), (\mathsf{pt}_B, p)\right)$, which is connected because Σ preserves connectedness.

We claim $\delta \circ \alpha \sim \mathsf{id}$. By the SIP, an identity $(B_1, F_1, p_2) = (B_2, F_2, p_3)$ amounts to a tuple consisting of $e : B_1 \simeq_* B_2$, $t : (b : B_1) \to F_1(b) \simeq F_2(e(b))$, and a coherence field $c : \mathsf{transp}^{F_2}(\mathsf{pt}_e, t(\mathsf{pt}_{B_1}, p_2)) = p_3$. Now, let (B, F, p) be a suitable triple. We have a composite e of pointed equivalences: $\|(b : B) \times F(b)\|_{n-1} \simeq_* \|(b : B) \times \|F(b)\|_{n-1}\|_{n-1} \simeq_* \|B\|_{n-1} \simeq_* B$. Note that e fits into the commuting triangle



This yields $t(b, y) : \mathsf{fib}_{|-|_{n-1}}(|(b, y)|_{n-1}) \simeq F(y)$ for $(b, y) : (b : B) \times F(b)$. As $\mathsf{transp}^F(\mathsf{pt}_e, t((\mathsf{pt}_B, p), ((\mathsf{pt}_B, p), \mathsf{refl}))) \equiv p$, we see that $\delta(\alpha(B, F, p)) = (B, F, p)$.

Next, we claim $\alpha \circ \delta \sim \mathrm{id}$. Let G be an n-group. Define $e : G \to (b : \|G\|_{n-1}) \times \mathsf{fib}_{|-|_{n-1}}(b)$ by $e(g) :\equiv \big(|g|_{n-1}, (g, \mathsf{refl})\big)$. It is easy to check that e is a pointed equivalence. By the SIP for pointed types, it follows that $\alpha(\delta(G)) = G$. ◻

Let G be an n-group. In accordance with [7, Section 4.3], we define a *G-module* as a family $H : G \to \mathbf{Ab}$ of abelian groups, which encodes an action of $\Omega(G)$ on $H(\mathsf{pt}_G)$ (in the sense of a map into $\mathrm{Aut}(H(\mathsf{pt}_G))$). The *group cohomology $H^m(G, H)$ of G over H* is the singular cohomology of G with coefficients in H, i.e., $H^m(G, H) :\equiv \|(u : G) \to_* K(H(u), m)\|_0$. Here, $(u : G) \to_* K(H(u), m)$ denotes the type of *pointed sections* of $K(H(-), m)$: a section $f : (u : BG) \to K(H(u), m)$ of the underlying type family together with a proof that f preserves the basepoint. Its elements are also called *cocycles*, a term from cohomology theory. A triple of the form (G, H, κ) with $\kappa : H^{n+2}(G, H)$ is called a *Sính n-triple*. When $n = 1$, we just call it a *Sính triple*. An *untruncated Sính n-triple* is a triple (G, H, κ) where $\kappa : (u : G) \to_* K(H(u), n + 2)$.

Theorem 9.2 (NGrp-Sinh-≃). *For $n \geq 1$, the type of $(n+1)$-groups is equivalent to that of untruncated Sính n-triples.*

Proof. It is more illuminating to start from the untruncated Sính n-triples. Let $G : \mathcal{U}_*^{\geq 0, \leq n}$. By Lemma 9.1, it suffices to show that $(H : G \to \mathbf{Ab}) \times ((u : G) \to_* K(H(u), n + 2))$ is equivalent to $\big(X : G \to \mathcal{U}^{\geq n, \leq n+1}\big) \times X(\mathsf{pt}_G)$.

For each $H : G \to \mathbf{Ab}$, we use Section 8 to recast the type of $(n+2)$-dimensional cocycles on G over H. We have

$$\begin{aligned}
&(u : G) \to_* K(H(u), n + 2) \\
\simeq\ &(u : G) \to_* T(K(H(u), n + 1)) \qquad \text{(by Example 8.3)} \\
\simeq\ &(X : G \to \mathcal{U}) \\
&\quad \times (d : (u : G) \to \|X(u)\| \\
&\quad \times ((x : X(u)) \to K(H(u), n + 1) \simeq_* (X(u), x))) \\
&\quad \times \big(X(\mathsf{pt}_G), d(\mathsf{pt}(G))\big) = \tau_c(K(H(u), n + 2))
\end{aligned}$$

By Theorem 8.2(1), $\big(X(\mathsf{pt}_G), d(\mathsf{pt}(G))\big) = \tau_c(K(H(u), n + 2))$ is equivalent to $X(\mathsf{pt}_G)$. By Corollary 8.5, $K(H(u), n + 1) \simeq_* (X(u), x)$ is equivalent to $(X(u)$ is n-connected$) \times (X(u)$ is $n+1$-truncated$) \times (\pi_{n+1}(X(u), x) = H(u))$. Therefore, after rearranging types, we see that $(H : G \to \mathbf{Ab}) \times ((u : G) \to_* K(H(u), n + 2))$ is equivalent to $(X : G \to \mathcal{U}) \times X(\mathsf{pt}_G) \times ((u : G) \to \|X(u)\| \times E_1(u) \times E_2(u))$—we label this latter type as Λ_G. Here, we have defined

$$E_1(u) :\equiv X(u) \to (X(u) \text{ is } n\text{-connected}, (n+1)\text{-truncated})$$
$$E_2(u) :\equiv (H : \mathbf{Ab}) \times ((x : X(u)) \to \pi_{n+1}(X(u), x) = H(u))$$

Let $u : G$. First, we observe that $\|X(u)\| \times E_1(u)$ holds if and only if $X(u)$ is n-connected and $(n+1)$-truncated. Next, we claim that $E_2(u)$ is contractible as soon as $X(u)$ is n-connected. Indeed, $E_2(u)$ is the type of diagonal fillers of

$$\begin{array}{ccc}
X(u) & \xrightarrow{\pi_{n+1}(X(u), -)} & \mathbf{Ab} \\
\downarrow & & \downarrow \\
\mathbf{1} & \longrightarrow & \mathbf{1}
\end{array}$$

Note that $\mathbf{Ab}$ is a 1-type, hence an n-type. If $X(u)$ is n-connected, the type of such fillers is contractible by virtue of the (n-connected, n-truncated) factorization system [25].

It follows that Λ_G is equivalent to $\big(X : G \to \mathcal{U}^{\geq n, \leq n+1}\big) \times X(\mathsf{pt}_G)$, which finishes the proof. ◻

Note 9.3 (Sinh-action). Let G be an $(n+1)$-group and let $(\Gamma_\sigma(G), H_\sigma(G), \kappa_\sigma(G))$ be the untruncated Sính n-triple produced by Theorem 9.2. We have that $\Gamma_\sigma(G) \equiv \|G\|_n$, known as the fundamental n-group $\Pi_n(G)$ of G. We claim that $H_\sigma(G)$ is the canonical action $C_{n,G}$ of $\Pi_n(G)$ on $\pi_{n+1}(G)$, with $C_{n,G}(|x|_n) :\equiv \pi_{n+1}(G, x)$.

Indeed, the proof of Theorem 9.2 tells us that $H_\sigma(G, u) = \pi_{n+1}(\mathsf{fib}_{|-|_n}(u), y)$ for all $u : \|G\|_n$ and $y : \mathsf{fib}_{|-|_n}(u)$. As $\mathbf{Ab}$ is a 1-type, it suffices to show that $\pi_{n+1}(\mathsf{fib}_{|-|_n}(|x|_n), (x, \mathsf{refl})) = \pi_{n+1}(G, x)$ for all $x : G$. We do so by induction on n using basic properties of path types of a truncation [Ω^'-hfib-Trunc]. It also follows easily from the long exact sequence [29, Theorem 8.4.6] for the fiber sequence $\mathsf{fib}_{|-|_n}(|x|_n) \to G \to \|G\|_n$.

We turn to a type-theoretic version of MacLane and Whitehead's classical bijection, extended to all dimensions ≥ 2.

Theorem 9.4 (Sinh-classif-set). *For $n \geq 1$, the components of $(n+1)$-groups are equivalent to those of Sính n-triples:* $\|\mathcal{U}_*^{\geq 0, \leq n+1}\|_0 \simeq \Big\|\big(G : \mathcal{U}_*^{\geq 0, \leq n}\big) \times (H : G \to \mathbf{Ab}) \times H^{n+2}(G, H)\Big\|_0$.

Proof. After applying $\|-\|_0$ to Theorem 9.2, we get the desired equivalence from the general interaction between truncation and Σ-types [29, Theorem 7.3.9]. ◻

By composing the equivalence $\mathrm{Ob}(\mathbf{c2Grp}) \simeq \mathrm{Ob}(\mathbf{2Type}_0^*)$ obtained from Theorem 6.11 with Theorems 9.2 and 9.4, we get the following characterizations of coherent 2-groups.

Theorem 9.5 (Type-equiv-main). *The type of coherent 2-groups is equivalent to the type of untruncated Sính triples, and the components of coherent 2-groups are equivalent to the components of Sính triples.*

10 Conclusion and open questions

Working in HoTT, we gave two algebraic classifications of pointed connected 2-types–the types corresponding to 2-groups under the homotopy hypothesis. The first was a (bi-adjoint) biequivalence between such types and coherent 2-groups (defined as monoidal groupoids with inverses). From this biequivalence we produced a path between these bicategories via univalence. The second classification was a type equivalence between pointed connected 2-types and Sính triples (which are defined via group cohomology). Our proof of this equivalence extended to n-groups for all $n \geq 2$.

Our work raises some open questions. First, the infinite loop space of an abelian group G is built from $K(G, 1)$ with suspensions and truncations [17, Section 5]. Can we build the double delooping of a braided 2-group and the infinite loop space of a symmetric 2-group from $\mathcal{K}_2$ in a similar way? We then would seek a tractable recursion principle for the higher deloopings to build—as asserted by the homotopy hypothesis—biequivalences between braided 2-groups and pointed 1-connected 3-types and between symmetric 2-groups and pointed n-connected $(n+2)$-types for all $n \geq 2$.

On the Sính-triple side, we have not classified maps of (untruncated) Sính triples. We define such a map $(G_1, H_1, \kappa_1) \to (G_2, H_2, \kappa_2)$ as a triple consisting of $\theta_1 : G_1 \to_* G_2$, $\theta_2 : (u : G_1) \to H_1(u) \to_{\mathbf{Ab}} H_2(\theta_1(u))$, and $\psi : K(\theta_2(-), 3) \circ \kappa_1 \sim_* \kappa_2 \circ \theta_1$ where $\sim_*$ is the type of pointed homotopies between pointed sections. Such maps form a 1-type since the type of ψ is 1-truncated by [7, Theorem 4.2]. Can we complete the bicategorical structure on the Sính triples? If so, can we promote Theorem 9.2 to a biequivalence?

Acknowledgments

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. We thank Favonia for invaluable comments on the writing of this paper.

References

[1] Benedikt Ahrens, Dan Frumin, Marco Maggesi, Niccolò Veltri, and Niels van der Weide. 2021. Bicategories in univalent foundations. *Mathematical Structures in Computer Science* 31, 10 (2021), 1232–1269. doi:10.1017/S0960129522000032

[2] Carlo Angiuli, Guillaume Brunerie, Thierry Coquand, Robert Harper, Kuen-Bang Hou (Favonia), and Daniel R. Licata. 2021. Syntax and models of Cartesian cubical type theory. *Mathematical Structures in Computer Science* 31, 4 (2021), 424–468. doi:10.1017/S0960129521000347

[3] John C. Baez and James Dolan. 1998. Categorification. In *Higher Category Theory (Contemporary Mathematics, Vol. 230)*, Ezra Getzler and Mikhail Kapranov (Eds.). American Mathematical Society, Providence, RI, 1–36. https://arxiv.org/abs/math/9802029

[4] John C. Baez and Aaron D. Lauda. 2004. Higher-dimensional algebra. V: 2-Groups. *Theory and Applications of Categories* 12 (2004), 423–491. http://eudml.org/doc/124217

[5] John C. Baez, Danny Stevenson, Alissa S. Crans, and Urs Schreiber. 2007. From loop groups to 2-groups. *Homology, Homotopy and Applications* 9, 2 (2007), 101 – 135.

[6] Tim Baumann. 2018. *The cup product on cohomology groups in Homotopy Type Theory.* Master's thesis. University of Augsburg. https://timjb.gitlab.io/masters-thesis/cp-in-hott.pdf

[7] Ulrik Buchholtz, Floris van Doorn, and Egbert Rijke. 2018. Higher Groups in Homotopy Type Theory. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '18)*. Association for Computing Machinery, New York, NY, USA, 205–214. doi:10.1145/3209108.3209150

[8] Ulrik Buchholtz and Johannes Schipp von Branitz. 2021. Using Displayed Univalent Graphs to Formalize Higher Groups in Univalent Foundations. (2021). https://ulrikbuchholtz.dk/durgs.pdf draft version.

[9] Pierre Cagne, Ulrik Buchholtz, Nicolai Kraus, and Marc Bezem. 2024. On symmetries of spheres in univalent foundations. arXiv:2401.15037 [cs.LO]

[10] Josep Elgueta. 2014. Permutation 2-groups I: Structure and splitness. *Advances in Mathematics* 258 (2014), 286–350. doi:10.1016/j.aim.2014.03.011

[11] Nick Gurski. 2012. Biequivalences in tricategories. *Theory and Applications of Categories* 26, 14 (2012), 349–384. http://www.tac.mta.ca/tac/volumes/26/14/26-14.pdf

[12] Perry Hart. 2026. Mechanized classifications of 2-groups. https://github.com/PHart3/2-groups-agda/tree/preprint Agda code (see the root README for an overview of the codebase).

[13] Pieter Hofstra and Martti Karvonen. 2024. Inner autoequivalences in general and those of monoidal categories in particular. *Journal of Pure and Applied Algebra* 228, 11 (2024), 107717. doi:10.1016/j.jpaa.2024.107717

[14] J. R. Isbell and F. B. Wright. 1966. Another equivalent form of the axiom of choice. *Proceedings of the American Mathematical Society* 17, 1 (1966), 174–174. https://www.ams.org/journals/proc/1966-017-01/S0002-9939-1966-0186535-8/S0002-9939-1966-0186535-8.pdf

[15] David Jaz Myers and Zyad Yasser. 2025. Higher Schreier Theory in Cubical Agda. *The Journal of Symbolic Logic* (2025), 1–17. doi:10.1017/jsl.2024.78

[16] Niles Johnson and Donald Yau. 2021. *2-Dimensional Categories.* Oxford University Press, Oxford. doi:10.1093/oso/9780198871378.001.0001

[17] Daniel R. Licata and Eric Finster. 2014. 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)* (Vienna, Austria) *(CSL-LICS '14)*. Association for Computing Machinery, New York, NY, USA, Article 66, 9 pages. doi:10.1145/2603088.2603153

[18] Peter Lefanu Lumsdaine and Michael Shulman. 2020. Semantics of higher inductive types. *Mathematical Proceedings of the Cambridge Philosophical Society* 169, 1 (2020), 159–208. doi:10.1017/S030500411900015X

[19] Saunders MacLane and J. H. C. Whitehead. 1950. On the 3-Type of a Complex. *Proceedings of the National Academy of Sciences* 36, 1 (1950), 41–48. doi:10.1073/pnas.36.1.41 arXiv:https://www.pnas.org/doi/pdf/10.1073/pnas.36.1.41

[20] Samuel Mimram and Emile Oleon. 2024. Delooping cyclic groups with lens spaces in homotopy type theory. In *Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science* (Tallinn, Estonia) *(LICS '24)*. Association for Computing Machinery, New York, NY, USA, Article 56, 15 pages. doi:10.1145/3661814.3662077

[21] Anders Mörtberg and Loïc Pujet. 2020. Cubical synthetic homotopy theory. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs* (New Orleans, LA, USA) *(CPP 2020)*. Association for Computing Machinery, New York, NY, USA, 158–171. doi:10.1145/3372885.3373825

[22] nLab authors. 2025. bigroupoid. https://ncatlab.org/nlab/show/bigroupoid. Revision 5.

[23] Behrang Noohi. 2007. Notes on 2-Groupoids, 2-Groups and Crossed Modules. *Homology, Homotopy and Applications* 9, 1 (2007), 75 – 106.

[24] Egbert Rijke. 2022. Introduction to Homotopy Type Theory. arXiv:2212.11082 [math.LO]

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

[26] Michael Shulman. 2019. All $(\infty, 1)$-toposes have strict univalent universes. arXiv:1904.07004 [math.AT]

[27] Andrew W Swan. 2022. On the Nielsen-Schreier Theorem in Homotopy Type Theory. *Logical Methods in Computer Science* Volume 18, Issue 1, Article 18 (Jan 2022), 15 pages. doi:10.46298/lmcs-18(1:18)2022

[28] Andrew W Swan. 2025. Computable and non-computable 2-groups in HoTT. In *Workshop on Homotopy Type Theory / Univalent Foundations 2025* (Genoa, Italy). https://hott-uf.github.io/2025/abstracts/HoTTUF_2025_paper_13.pdf

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

[30] Niccolò Veltri and Niels van der Weide. 2021. Constructing Higher Inductive Types as Groupoid Quotients. *Logical Methods in Computer Science* Volume 17, Issue 2, Article 8 (Apr 2021), 42 pages. doi:10.23638/LMCS-17(2:8)2021

[31] Andrea Vezzosi, Anders Mörtberg, and Andreas Abel. 2021. Cubical Agda: A dependently typed programming language with univalence and higher inductive types. *Journal of Functional Programming* 31 (2021), e8. doi:10.1017/S0956796821000034

[32] David Wärn. 2023. Eilenberg–Maclane spaces and stabilisation in homotopy type theory. *Journal of Homotopy and Related Structures* 18, 2 (21 Sep 2023), 357–368. doi:10.1007/s40062-023-00330-5

A The short definition of 2-group morphism

Let G_1 and G_2 be 2-groups and $f_0 : G_1 \to G_2$ be a function between their underlying types. We prove that the function (∂), found in Note 4.6, is an equivalence of types. This amounts to showing that if f_0 has the data D_s making up the short definition of a 2-group morphism, then it also has unique preservation data for id, which we call P_u, and unique preservation data for $(-)^{-1}$, which we call P_i.[1]

Suppose that f_0 has the data D_s. The data P_u consists of a path $u : \mathsf{id} = f_0(\mathsf{id})$ and commuting diagrams for each $x : G_1$:

$$\begin{array}{ccc} f_0(x) & \overset{\rho(f_0(x))}{=\!=\!=} & f_0(x) \otimes \mathsf{id} \\ \mathsf{ap}_{f_0}(\rho(x)) \Vert & r_u(x) & \Vert \mathsf{ap}_{f_0(x)\otimes -}(u) \\ f_0(x \otimes \mathsf{id}) & \underset{\mu_{x,\mathsf{id}}}{=\!=\!=} & f_0(x) \otimes f_0(\mathsf{id}) \end{array}$$

$$\begin{array}{ccc} f_0(x) & \overset{\lambda(f_0(x))}{=\!=\!=} & \mathsf{id} \otimes f_0(x) \\ \mathsf{ap}_{f_0}(\lambda(x)) \Vert & \ell_u(x) & \Vert \mathsf{ap}_{-\otimes f_0(x)}(u) \\ f_0(\mathsf{id} \otimes x) & \underset{\mu_{\mathsf{id},x}}{=\!=\!=} & f_0(\mathsf{id}) \otimes f_0(x) \end{array}$$

[1] The only reference we have found mentioning that it's possible to recover P_u and P_i simultaneously is [10, Section 2.3]. The author, however, leaves the proof to the reader.

The data P_i consists of a path $i_x : f_0(x)^{-1} = f_0(x^{-1})$ and commuting diagrams for each $x : G_1$:

$$\begin{array}{ccccc} f_0(x) \otimes f_0(x)^{-1} & \overset{\mathsf{ap}_{f_0(x)\otimes -}(i_x)}{=\!=\!=} & f_0(x) \otimes f_0(x^{-1}) & \overset{\mu_{x,x^{-1}}}{=\!=\!=} & f_0(x \otimes x^{-1}) \\ \Vert\ \mathsf{rinv}(f_0(x))\ \Vert & & r_i(x) & & \Vert\ \mathsf{ap}_{f_0}(\mathsf{rinv}(x))\ \Vert \\ \mathsf{id} & & \underset{u}{=\!=\!=\!=\!=\!=} & & f_0(\mathsf{id}) \end{array}$$

$$\begin{array}{ccccc} f_0(x)^{-1} \otimes f_0(x) & \overset{\mathsf{ap}_{-\otimes f_0(x)}(i_x)}{=\!=\!=} & f_0(x^{-1}) \otimes f_0(x) & \overset{\mu_{x^{-1},x}}{=\!=\!=} & f_0(x^{-1} \otimes x) \\ \Vert\ \mathsf{linv}(f_0(x))\ \Vert & & \ell_i(x) & & \Vert\ \mathsf{ap}_{f_0}(\mathsf{linv}(x))\ \Vert \\ \mathsf{id} & & \underset{u}{=\!=\!=\!=\!=\!=} & & f_0(\mathsf{id}) \end{array}$$

Both $f_0(x) \otimes - : G_2 \to G_2$ and $- \otimes f_0(x) : G_2 \to G_2$ are equivalences of types [mu-pre-iso *and* mu-post-iso]. Thus, we have a unique choice of u satisfying $r_u(\mathsf{id})$. Moreover, we have a unique choice i_{right} of i satisfying r_i and, separately, a unique choice i_{left} of i satisfying ℓ_i.

We first recover P_u. Let $x : G_1$. As ℓ_u and r_u are families of propositions, it suffices to prove our choice of u satisfies $\ell_u(x)$ and $r_u(x)$. First, we show that $r_u(\mathsf{id})$ implies $\ell_u(x)$ [rho-to-lam]. Second, we show that $\ell_u(x)$ implies $r_u(x)$ [lam-to-rho], so that $r_u(\mathsf{id})$ implies both $r_u(x)$ [rhoid-to-rho] and $\ell_u(x)$. The formal proofs of both steps give the details with explicit equational reasoning, which largely matches a pen-and-paper proof thanks to Agda's instance search.

It remains to recover P_i. To do so, we take i_{right} and show that it satisfies ℓ_i in the presence of P_u, which we have already recovered. (We could switch the roles of r_i and ℓ_i.) We refer the reader to either our mechanized proof [rinv-to-linv] or [4, Theorem 6.1] for the details.

Remark A.1. By keeping track of indices of hom-types, it's easy to extend our proof to pseudofunctors of (locally univalent) *bigroupoids* [22]. (A 2-group is a single-object bigroupoid.) This means that a pseudofunctor $F : \mathcal{B} \to \mathcal{C}$ of bigroupoids is simply a 2-*semifunctor* [6, Chapter 5], which consists of a function $F_0 : \mathsf{Ob}(\mathcal{B}) \to \mathsf{Ob}(\mathcal{C})$, an action $F_1 : \mathsf{hom}_{\mathcal{B}}(a, b) \to \mathsf{hom}_{\mathcal{C}}(F_0(a), F_0(b))$ on 1-cells, and a family of 2-cells $F_c(f, g) : F_1(g \circ f) = F_1(g) \circ F_1(f)$ that respects the associator.

B Coherence conditions for Step 1 of Theorem 6.11

We verify that ξ_1 satisfies the relevant coherence conditions. In the case of unitors, we want to prove that the following square commutes for each pointed connected 2-type X:

$$\begin{array}{ccc} \mathsf{id}_X \circ \xi_0(X) & \overset{\xi_1(\mathsf{id}_X)}{=\!=\!=\!=} & \xi_0(X) \circ \mathcal{K}_2(\Omega(\mathsf{id}_X)) \\ \textit{left unitor} \Big\| & & \Big\| \textit{composite } \mathsf{id} \textit{ preservation} \\ \xi_0(X) & \underset{\textit{right unitor}}{=\!=\!=\!=} & \xi_0(X) \circ \mathsf{id}_{\mathcal{K}_2(\Omega(X))} \end{array}$$

By the SIP for pointed homotopies, this amounts to a homotopy $M_1(X)$ between the homotopies underlying the 2-cells in the square along with a dependent path $M_2(X)$ over $M_1(X)$ between the corresponding proofs of pointedness. We define $M_1(X)$ by applying Lemma 6.5 to $\mathsf{base}^{\sim\sim} :\equiv \mathsf{refl}$ and $\mathsf{loop}^{\sim\sim} :\equiv M_1(X)\text{-}\,\mathsf{loop}$. Here, for each loop $p : x_0 = x_0$, $M_1(X)\text{-}\,\mathsf{loop}(p)$ is a path between two homotopy-naturality squares at $\mathsf{loop}(p)$—call them *NatSq₁* and *NatSq₂*, as in Fig. 10. Each of the upper three (hence bottom three) paths of the upper square in Fig. 10 reduces to refl by the base computation rule of $\mathcal{K}_2$-induction. To build $M_1(X)\text{-}\,\mathsf{loop}(p)$, we use Lemma 3.2 to decompose *NatSq₁* into three paths $L_1(p)$, $L_2(p)$, and $L_3(p)$ corresponding to the three homotopy naturality sub-squares shown in Fig. 10, from left to right [KLoop-ptr-idf-aux1]. By the typal β-rule of Lemma 6.4, these paths fit into the trio of commuting diagrams in Fig. 11 (which are mechanized at [KLoop-ptr-idf-aux0]). We further adjust $L_1(p)$ by rewriting the middle path

$$\mathsf{ap}_{\mathsf{id}_{\Omega(X)}}(p) = \mathsf{ap}_{\mathsf{fun}(\xi_0(X))}(\mathsf{loop}(\mathsf{ap}_{\mathsf{id}_{\Omega(X)}}(p)))$$

of $H_1(f)\text{-}\,\mathsf{loop}(p)$ via homotopy naturality as in Fig. 12. Now, notice that *NatSq₂* is trivial. Thus, we can derive $M_1(X)\text{-}\,\mathsf{loop}(p)$ by proving the composition of $L_1(p)$, $L_2(p)$, and $L_3(p)$ is trivial. We do so by repeatedly cancelling point β-rules [KLoop-ptr-idf-coher]. Finally, our definition of $M_1(X)$ makes it trivial to define $M_2(X)$, thereby completing the coherence with the unitors [KLoop-coher-unit]. The coherence with the associator is similar but more complicated [KLoop-PT-assoc].

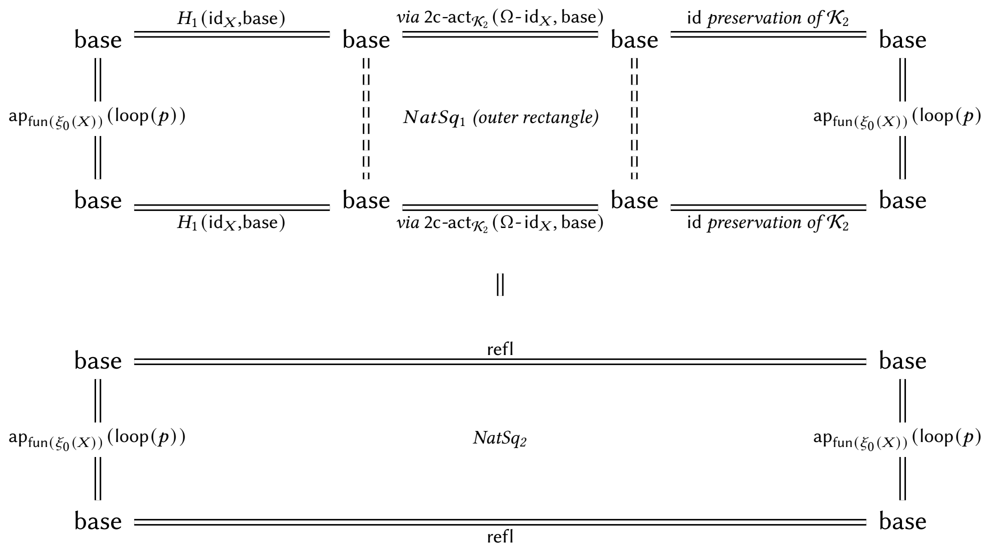


Figure 10: type of $M_1(X)$- $\mathsf{loop}(p)$, where $\mathsf{2c\text{-}act}_{\mathcal{K}_2}$ is as in Example 6.6 (Note: $NatSq_1$ fills the *outer* diagram.)

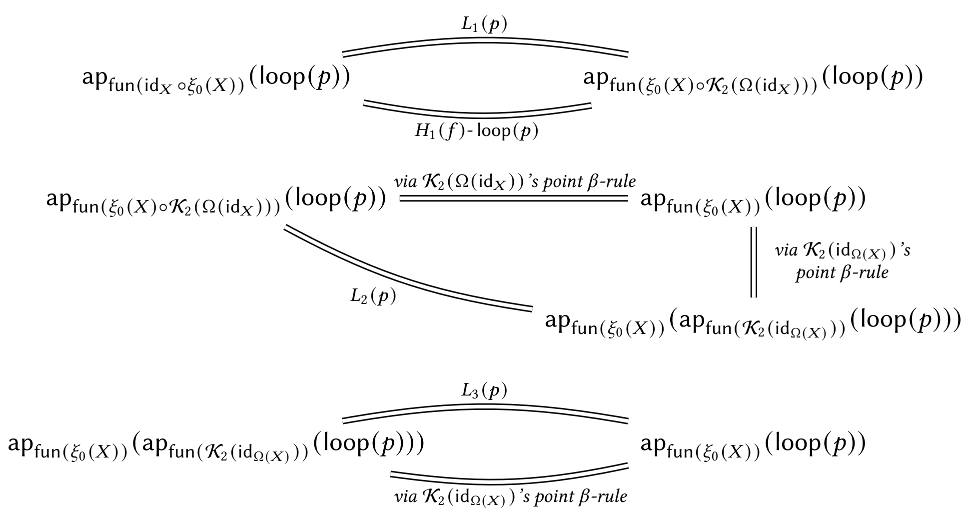


Figure 11: rewriting $L_1(p)$, $L_2(p)$, and $L_3(p)$, respectively

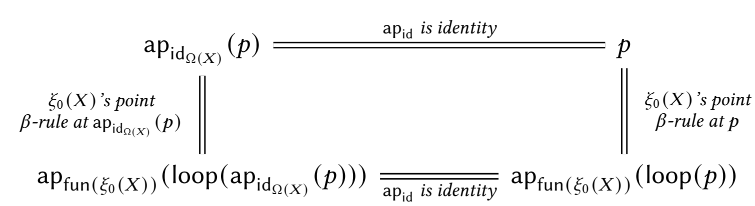


Figure 12: rewriting the middle path of $H_1(f)$- $\mathsf{loop}(p)$