A mechanized characterization of coherent 2-groups

Perry Hart

University of Minnesota, Twin Cities

SumTopo 2025

Plan

1. Motivation and background
2. Working in *homotopy type theory*, generalize the categorical equivalence $\mathbf{Grp} \simeq \mathbf{Ptd}^{conn}_{\leq 1}$ to the case of 2-groups.

   Result: *biequivalence between coherent* 2*-groups and pointed connected* 2*-types.*
3. Use the univalence axiom to get an *identity* between the $(2,1)$-category of coherent 2-groups and that of pointed connected 2-types.
4. Brief look at the Agda mechanization

Motivation

- 2-groups and pointed connected 2-truncated spaces have rich histories in algebraic topology.
- The biequivalence has been suggested previously in both the classical and the type-theoretic setting.
- Homotopy type theory lets us prove the biequivalence in a *constructive* system with highly general semantics.

Dependent type theory

Martin-Löf type theory (MLTT) takes *types* and *type inhabitation* as primitive notions.

We try to prove judgments of the form

$$\Gamma \vdash t : X \qquad \textit{t has type X in context } \Gamma$$

Think of types as formulas and terms as proofs of them.

Types can depend on terms: $P(x)$ *is a type depending on a term* x*.*

A few important type formers in MLTT:

- *the type* $X \to Y$ *of functions from type* X *to type* Y
- $(x : X) \to P(x)$ *for all* x *in* X*, we have an element of* $P(x)$*.*
- For all $x, y : X$, the *identity type* $x = y$, inductively generated by reflexivity $\text{refl}_x : x = x$.

Homotopy type theory (HoTT)

HoTT is a variant of dependent type theory where

- types are interpreted as *homotopy types* (as spaces up to homotopy, if you like).

  interpretation in all $(\infty, 1)$*-toposes*

  Elements of identity types are called ***paths****.*

- proof-checking

  does term t have type X in context Γ?

  is (at least) semi-decidable, with implementations in the proof assistants Agda and Rocq.

  If Agda returns successfully, then we're good.

HoTT extends MLTT with at least one of

- **the univalence axiom**

  Identity of types is the same as equivalence of types.

- **higher inductive types**.

  Generalize inductive types by allowing generators of id types.

  Directly build and study higher-dimensional spaces.

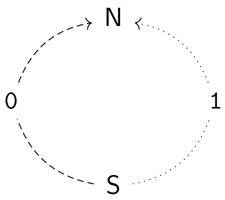
Higher inductive types (HITs)

New syntax for defining types

- Generated by both points and paths.
- Give us nontrivial (i.e., non-refl) elements of $x = x$.
- Similarly to ordinary inductive types, HITs come with induction/recursion principles.

Example (the circle S^1)

Generated by two points N, S and two identities $0, 1 : \mathsf{S} = \mathsf{N}$:



A critical notion

An *n-**type***, or *n-truncated type*, is a type whose identity types are trivial above level n.

Examples

- A type X is a 0-type, or a *set*, if each of its loop spaces $x = x$ contains only refl_x.
- A type X is a 1-*type* if all of its loop spaces are sets.
- A type X is a 2-*type* if all of its loop spaces are 1-types.

The 1-dimensional case

1. The *delooping* of a group G (Licata and Finster):

   the 1-truncated HIT[1] $K(G,1)$ generated by

   - a point base : $K(G,1)$
   - a homomorphism loop : $G \to \Omega(K(G,1), \text{base})$:

   $$\begin{aligned} \text{loop}_0 &: G \to \text{base} = \text{base} \\ \text{loop}_1 &: (x, y : G) \to \text{loop}_0(x \otimes y) = \text{loop}_0(x) \cdot \text{loop}_0(y) \end{aligned}$$

   Theorem: loop is an isomorphism.

2. Equivalence of categories (Buchholtz, van Doorn, and Rijke):

   $$\mathbf{Grp} \underset{\Omega}{\overset{K(-,1)}{\rightleftarrows}} \mathbf{Ptd}^{conn}_{\leq 1}$$

[1] It is permitted to truncate a HIT at level n and thereby make it an n-type.

2-groups

A *(coherent) 2-group* (Baez and Lauda) is a 1-type G with

- a neutral element e
- a binary operation $\otimes : G \to G \to G$, called the *tensor product*
- a right unitor ρ, a left unitor λ, and an associator α for $\otimes$

$$\textit{e.g.,} \quad \alpha \;:\; (x, y, z : X) \to (x \otimes y) \otimes z = x \otimes (y \otimes z)$$

- a triangle identity and a pentagon identity
- an *inverse* operation $(-)^{-1} : G \to G$
- paths $\mathrm{linv}_x : x^{-1} \otimes x = \mathrm{id}$ and $\mathrm{rinv}_x : x \otimes x^{-1} = \mathrm{id}$ for each $x : G$ such that linv and rinv satisfy two zig-zag identities.

Example

For every pointed 2-type X, the loop space $\Omega(X)$ equipped with path concatenation has the structure of a 2-group.

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 $\mu_{x,y} : f_0(x) \otimes f_0(y) = f_0(x \otimes y)$ that respects the associator.[2]

Justification for our short definition of 2-group morphism:

For each function $f_0 : G_1 \to G_2$ between the underlying types of 2-groups, the forgetful function

$$\textit{fully explicit notion on } f_0 \quad \to \quad \textit{short notion on } f_0$$

is an equivalence.

[2] Intuitively, this means you can get from the associator in G_1 to the associator in G_2 by following μ.

Delooping a 2-group

Let G be a 2-group.

Construct its delooping by generalizing the delooping $K(-,1)$ of an ordinary group.

Form the 2-truncated HIT $K_2(G)$ generated by

- a point base : $K_2(G)$
- a morphism of 2-groups loop : $G \to \Omega(K_2(G), \mathsf{base})$

Key feature: no path constructors for units or inverses.

makes induction on $K_2(G)$ much more tractable

Theorem: loop is an isomorphism.

Bicategories

For us, *bicategory* means $(2,1)$-category whose 2-cells (morphisms between morphisms) are paths.

That is, a *bicategory* consists of a type Ob of objects together with

- a doubly indexed family hom of 1-types over Ob
- a composition operation
  $\circ : \mathrm{hom}(b, c) \to \mathrm{hom}(a, b) \to \mathrm{hom}(a, c)$ for all $a, b, c : \mathrm{Ob}$
- an identity morphism id_a for each $a : \mathrm{Ob}$ together with two 2-cells (i.e., paths between morphisms): the *right unitor* and the *left unitor*
- an *associator* 2-cell satisfying both the triangle identity with the unitors and the pentagon identity.

Let $\mathcal{C}$ and $\mathcal{D}$ be bicategories.

A *pseudofunctor from* $\mathcal{C}$ *to* $\mathcal{D}$ is a function $F_0 : \mathrm{Ob}(\mathcal{C}) \to \mathrm{Ob}(\mathcal{D})$ together with

- a function $F_1 : \mathrm{hom}_{\mathcal{C}}(a, b) \to \mathrm{hom}_{\mathcal{D}}(F_0(a), F_0(b))$ for all $a, b : \mathrm{Ob}$
- a 2-cell $F_{\mathrm{id}}(a) : F_1(\mathrm{id}_a) = \mathrm{id}_{F_0(a)}$ for each $a : \mathrm{Ob}$
- 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 witnessing that F_{id} and $F_\circ$ commute with the right unitors, with the left unitors, and with the associators.

Let $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{C}$ be pseudofunctors.

A *pseudotransformation from F to G* consists of

- a component morphism $\eta_0(a) : F_0(a) \to G_0(a)$ for each $a : \mathrm{Ob}(\mathcal{C})$
- a 2-cell $\eta_1(f)$ making the square

$$\begin{array}{ccc} F_0(a) & \xrightarrow{F_1(f)} & F_0(b) \\ {\scriptstyle \eta_0(a)}\downarrow & & \downarrow{\scriptstyle \eta_0(b)} \\ G_0(a) & \xrightarrow[G_1(f)]{} & G_0(b) \end{array}$$

  commute for all $a, b : \mathrm{Ob}(\mathcal{C})$ and $f : \hom_{\mathcal{C}}(a, b)$.
- a coherence identity witnessing that η_1 commutes with the unitors and one witnessing that it commutes with the associators.

The type of such pseudotransformations is denoted by $F \Rightarrow G$.

The biequivalence

A *biequivalence between* $\mathcal{C}$ *and* $\mathcal{D}$ is a pseudofunctor $F : \mathcal{C} \to \mathcal{D}$ together with

- a pseudofunctor $G : \mathcal{D} \to \mathcal{C}$
- a pseudotransformation $\tau_1 : F \circ G \Rightarrow \mathrm{id}_{\mathcal{D}}$ each of whose components is an adjoint equivalence[3] in $\mathcal{D}$
- a pseudotransformation $\tau_2 : \mathrm{id}_{\mathcal{C}} \Rightarrow G \circ F$ each of whose components is an adjoint equivalence in $\mathcal{C}$.

[3] An adjoint equivalence between objects in a bicategory resembles an adjoint equivalence between ordinary categories.

The biequivalence

A *biequivalence between* $\mathcal{C}$ *and* $\mathcal{D}$ is a pseudofunctor $F : \mathcal{C} \to \mathcal{D}$ together with

- a pseudofunctor $G : \mathcal{D} \to \mathcal{C}$
- a pseudotransformation $\tau_1 : F \circ G \Rightarrow \mathrm{id}_{\mathcal{D}}$ each of whose components is an adjoint equivalence[3] in $\mathcal{D}$
- a pseudotransformation $\tau_2 : \mathrm{id}_{\mathcal{C}} \Rightarrow G \circ F$ each of whose components is an adjoint equivalence in $\mathcal{C}$.

Theorem

We have a biequivalence

$$\mathbf{2Grp} \underset{\Omega}{\overset{K_2}{\rightleftarrows}} \mathbf{Ptd}^{conn}_{\leq 2}$$

[3] An adjoint equivalence between objects in a bicategory resembles an adjoint equivalence between ordinary categories.

A full-fledged identity

From the biequivalence, we extract a pseudofounctor $\Omega : \mathbf{Ptd}^{conn}_{\leq 2} \to \mathbf{2Grp}$ that is an equivalence on objects and homs.

By the univalence axiom, this data gives us an identity $\mathbf{Ptd}^{conn}_{\leq 2} = \mathbf{2Grp}$ in the type of all bicategories.

From this identity, it is trivial to prove that any bicategorical property of one holds for the other.

Agda mechanization

Resource usage: 167.5 minutes, 29 GB of memory

Induction on $K_2(G)$ produces large computations.

General tip:

Whenever possible, prevent Agda from unfolding large terms during type checking.

Conclusion

Takeaways:

- Coherent 2-groups are biequivalent to pointed connected 2-types.
- A fully mechanized, constructive proof inside HoTT.

Preprint: `https://phart3.github.io/2Grp-biequiv-preprint.pdf`

Agda code:
`https://github.com/PHart3/2-groups-agda`

Thanks!

References

John C. Baez, Aaron D. Lauda. 2004. *Higher-Dimensional Algebra V: 2-Groups*.

Ulrik Buchholtz, Floris van Doorn, and Egbert Rijke. 2018. *Higher Groups in Homotopy Type Theory*.

Daniel R. Licata, Eric Finster. 2014. *Eilenberg-MacLane spaces in homotopy type theory*.