# Using Displayed Univalent Graphs to Formalize Higher Groups in Univalent Foundations

Johannes Schipp von Branitz and Ulrik Buchholtz

October 13, 2021

**Abstract**

We introduce displayed univalent reflexive graphs, a natural analogue of displayed categories, as a framework for uniformly internalizing composite mathematical structures in homotopy or cubical type theory. This framework is then used to formalize the definitions of, and equivalence of, strict 2-groups and crossed modules. Lastly, foundations for the development of higher groups from the classifying space perspective in cubical type theory are laid. All results are formalized in Cubical Agda.

## 1 Introduction

We provide a framework for constructing identifications between composite types in univalent foundations. In univalent foundations, mathematical structures (such as groups, rings, categories, etc.) are represented as types, and the types are interpreted (informally and in models) as $\infty$-groupoids. The identity types of a type represent the ways in which elements of that type can be identified. From Voevodsky's univalence axiom it follows that the identity type $A =_{\mathrm{Type}} B$ in the universe, Type, for two types $A$ and $B$, is equivalent to the type of equivalences $A \simeq B$. It is a prototype for the *structure identity principle* (SIP), as described by Aczel [1]:

> *Equivalent mathematical structures are structurally identical.*

The meaning of "equivalence" depends on the kind of structure we're talking about. For groups and rings it is *isomorphism*, while for categories it is *equivalence of categories*, etc. Since type theory is a structural foundation, equivalent structures are structurally *indistinguishable*. The SIP goes one step further, and requires that equivalent structures $A$ and $B$ can be *identified*, so that any property or construction made for $A$ can be applied to $B$. This is exactly what is captured by the identity type and its elimination rule. Recently, the structure identity principle has also been called the *univalence principle* [4].

Here, we consider the problem of how to implement the SIP for complicated structures built out of simpler components. We describe the SIP for a type in terms of a univalent reflexive graph (URG) structure. To construct the SIP for a composite type, we take inspiration from work on displayed categories [3] and bicategories [2], and introduce the notion of a displayed univalent reflexive graph (DURG) structure. We apply this framework to the problem of defining crossed modules and strict 2-groups as URGs, and we then use the (D)URG structures in order to prove that the corresponding types are equivalent. An important benefit of this approach is that we can leverage transport of structure in the lower levels of the hierarchy in order to obtain equivalences higher up.

These particular structures (crossed modules and strict 2-groups) also have natural 1-category structures, and an alternative would have been to define the (displayed) 1-category structures instead. But not every mathematical structure has a 1-category structure. (Indeed, the type of objects of a univalent 1-category is 1-truncated. And some types of structure have no interesting category structure at all.) URGs apply to all structures, whether truncated or not. To illustrate this, we also construct the SIP for various kinds of higher groups.

Our framework of (D)URGs can be applied to any type $A$ to specify a convenient mathematical notion of equivalence between elements of $A$. This allows us to hide the specifics of the type theory in use and hence enables uniform reasoning across similar type theories. In fact, we can view a URG structure on $A$ as a truncated part of an $\infty$-groupoid structure on $A$. At the moment, we do not know whether it is possible to define the type of $\infty$-category structures on a type inside type theory. (It's possible in 2-level type theory [7]). In contrast, we *do* know how to define the type of $\infty$-groupoid structures on $A$: it is the contractible type. Likewise, the type of URG structures on a type is also contractible. So it seems like nothing is gained. What is gained is an *intensionally* different presentation of the identifications in a type that can be more useful. (Likewise, using a hypothetical definition of $(\infty, 1)$-category structures one could carve out the $\infty$-groupoid structures, and these could be intensionally different than the ones given by iterated identity types.)

We have implemented our results[1] in the *cubical* library [15] for Cubical Agda [18], which is a switch for Agda that implements cubical type theory [10, 12].

## Related work

Coquand and Danielsson [11] derived the SIP in Agda for a range of structures, including 1-truncated, first-order algebraic structures. Using first-order logic with dependent sorts (FOLDS), the SIP for more complicated, but purely relational, structures of finite truncation level has been developed [4]. (This work includes relational definitions of 1-categories, †-categories, and bicategories.)

Escardó [13] proposed a technique for deriving the SIP for structures on a type using the so-called *standard notion of structure* (SNS). This is a special case of a DURG structure where the base structure is the URG structure provided by the univalence axiom for the universe of types. A variation of this was previously implemented in the *cubical* library, as reported in [6].

We are the first to formalize the equivalence of types between crossed modules and strict 2-groups. Previously, von Raumer [19] formalized the related equivalence of precategories of generalized crossed modules and double groupoids [8] in the HoTT-mode for Lean 2. However, that work didn't establish the SIP, or equivalently, prove that these precategories are univalent.

Also, in the HoTT-mode for Lean 2, the equivalence between pointed connected 1-types and groups has previously been formalized [9]. The formalization

[1] https://github.com/Schippmunk/cubical

here is novel, because cubical type theory natively supports higher inductive types and the resulting arguments thus become vastly simpler.

### Outline

In Section 2 we define URGs and DURGs, and derive various constructions on these. Then in Section 3 we define hierarchies of structures building up to the definitions of strict 2-groups and crossed modules with natural notions of equivalence. In Section 4 we prove the SIP for various kinds of higher groups and we prove that 1-groups are equivalent to the usual axiomatic notion of a group. We conclude in Section 5 with an outlook towards future work.

## 2 Displayed Structures

A displayed category $D$ over a category $C$ is equivalent to a category $D'$ and a functor $D' \to C$, but instead of having a single collection of objects with a map to the objects of $C$, the objects are given as a family indexed by objects of $C$, and similarly for the morphisms. By taking the total category of a displayed category, the process of adding structure can be iterated.

The type-theoretical analogue, a displayed type $B$ over a type $A$, is simply a type family over $A$. Via fibrant replacement, this data is equivalent to that of a type $B'$ together with a function $B' \to A$. The total category corresponds to the $\Sigma$-type $(a : A) \times B\ a$.

The structure of a type is reflected by its identity type. Usually one works with an explicit representation – a characterization – of these identity types. For instance, instead of paths between groups one prefers isomorphisms. Therefore, instead of mere types, we use "displayed characterizations of identity types".

**Definition 2.1.** A *univalent reflexive graph (URG) structure* on a type $A$ is a triple[2] $\langle \cong_A, \rho_A, \mathrm{uni}_A \rangle$ consisting of a reflexive binary relation

$$\cong_A : A \to A \to \mathrm{Type}, \qquad \rho_A : (a : A) \to a \cong_A a,$$

and a witness of *univalence*, that is a proof that the natural map of type

$$(a = b) \to a \cong b,$$

which sends refl to $\rho_A$, is an equivalence for all $a, b : A$.

It is often convenient to use the following paraphrase of the fundamental theorem of identity types [17, Thm. 5.8.4] to construct URGs.

**Theorem 2.2.** *A reflexive graph* $\cong$ *on* $A$ *is univalent if and only if the* $\cong$-singleton $(b : A) \times a \cong b$ *is contractible for every* $a : A$.

Examples of URGs include observational equality on the natural numbers, equivalences in a univalent universe and univalent 1-precategories. Any type has a trivial URG structure given by the identity type. It is extensionally the only one: the type of *small* univalent graphs on $A$ is contractible.

[2] We leave the subscript and last two components implicit whenever appropriate.

**Definition 2.3.** Let $\cong_A$ be a URG structure on $A$. A *displayed univalent reflexive graph (DURG) structure* on $B : A \to \text{Type}$ is a triple[3] $\langle \cong_A^B, \rho_A^B, \text{uni}_A^B \rangle$ consisting of a *displayed relation* of type

$$\{a, a' : A\} \to B\ a \to a \cong_A a' \to B\ a' \to \text{Type}$$

and a reflexivity term of type

$$\{a : A\} \to (b : B\ a) \to b \cong_{\rho\, a}^B b',$$

such that the natural map

$$b =_{B\ a} b' \to b \cong_{\rho\, a}^B b'$$

is an equivalence for all $a : A$ and $b, b' : B\ a$.

Propositions are particularly easy to display, cf. [3, Ex. 3.6].

**Proposition 2.4.** *Let $P : A \to \text{Type}^{\leq -1}$ be a propositional family over a URG. Then $p \cong_q^P p' :\equiv \mathbf{1}$ defines a DURG structure on $P$.*

*Proof.* By Theorem 2.2, it suffices to show that the $\cong_{\rho\, a}^P$-singleton $(p' : P\ a) \times p \cong_{\rho\, a}^P p'$ is contractible for all $a : A$ and $p : P\ a$. By definition, that singleton is just $(p' : P\ a) \times \mathbf{1}$ and hence, equivalent to the inhabited proposition $P\ a$. ☐

As promised, a DURG structure on a type family induces a URG structure – a characterization of the identity types – on its total type, cf. [3, Thm. 7.4].

**Theorem 2.5.** *For any DURG structure $\cong_A^B$ on $B : A \to \text{Type}$ there is an associated* total URG $\int \cong_A^B$ *on $(a : A) \times B\ a$ with the relation*

$$\langle a, b \rangle \cong_\Sigma \langle a', b' \rangle :\equiv (p : a \cong_A a') \times b \cong_p^B b'.$$

*Proof.* Reflexivity follows from the reflexivity of the (fiberwise) relations on $A$ and $B$. The $\cong_\Sigma$-singleton

$$(\langle a', b' \rangle : (a' : A) \times B\ a') \times (p : a \cong_A a') \times b \cong_p^B b'$$

at $\langle a, b \rangle$ is equivalent to

$$(\langle a', p \rangle : (a' : A) \times a \cong_A a') \times (b' : B\ a') \times b \cong_p^B b',$$

which is a $\Sigma$-type of contractible types. ☐

**Lemma 2.6.** *If there are URG structures on types $A$ and $B$, we can display $B$ over $A$ via $b \cong_p b' :\equiv b \cong_B b'$. Taking the total space yields the* product URG *structure on $A \times B$.*

**Lemma 2.7.** *If $\cong_A^B$ and $\cong_A^C$ are DURG structures over the same type $A$, then $\cong_A^B$ can be lifted to be displayed over $\int \cong_A^C$ by ignoring the $C$-component:*

$$b \cong_{\langle \rho\, a, \rho\, c \rangle} b' :\equiv b \cong_{\rho\, a} b'.$$

[3] We leave the second and third component, and the sub- and superscript types implicit whenever appropriate. Thus, we write $b \cong_q^B b'$, or even $b \cong_q b'$, instead of $(\cong_A^B)\ b\ q\ b'$ for any $a, a' : A$, $q : a \cong_A a'$, $b : B\ a$, and $b' : B\ a'$.

In order to construct fiberwise equivalences between structures it is necessary to reassociate the towers of structure at hand.

**Theorem 2.8.** *Given a DURG structure on* $C : (a : A) \times B\ a \to \mathrm{Type}$ *over* $\int \cong^B_A$*, the displayed relation*

$$\langle b, c\rangle \cong_p \langle b', c'\rangle \ :\equiv\ (q : b \cong_p b') \times c \cong_{\langle p,q\rangle} c'$$

*defines a DURG structure on* $\lambda\, a \mapsto (b : B\ a) \times C\langle a, b\rangle$*.*

*Proof.* Let $a : A$. The relational singleton

$$(\langle b', c'\rangle : (b' : B\ a) \times C\langle a, b'\rangle) \times (p : b \cong_{\rho\, a} b') \times c \cong_{\langle \rho\, a, p\rangle} c'$$

at $\langle b, c\rangle$ is equivalent to the contractible type

$$(\langle b', p\rangle : (b' : B\ a) \times b \cong_{\rho\, a} b') \times (c' : C\langle a, b'\rangle) \times c \cong_{\langle \rho\, a, p\rangle} c'. \qquad \square$$

**Corollary 2.9.** *Two DURG structures* $\cong^B_A$ *and* $\cong^C_A$ *with a common base structure induce a product DURG structure on* $\lambda\, a \mapsto B\ a \times C\ a$*.*

**Proposition 2.10.** *Any DURG structure* $\cong^B_A$ *induces a URG structure on* $(a : A) \to B\ a$ *with the relation*

$$f \cong_\Pi g \ :\equiv\ (a : A) \to f\ a \cong_{\rho\, a} g\ a.$$

*Proof.* Univalence of $\cong^B_A$ and function extensionality produce the desired equivalence

$$\big((a : A) \to f\ a \cong_{\rho\, a} g\ a\big) \simeq \big((a : A) \to f\ a = g\ a\big) \simeq (f = g). \qquad \square$$

Now, let us explore how to use URG structures to construct equivalences between types. Oftentimes, equivalences are constructed from quasi-inverse maps. The natural analog for types with URG structures is a relational isomorphism: A *relational isomorphism* of graphs $\langle A, \cong\rangle$ and $\langle A', \cong'\rangle$ consists of maps $f$ and $g$ back and forth, such that $g(f\ a) \cong a$ and $f(g\ a') \cong a'$ for all $a : A$ and $a' : A'$. If the graphs are univalent, such a relational isomorphism induces a quasi-isomorphism and hence an equivalence between the underlying types $A$ and $A'$.

To generalize this to displayed structures, we note that if $\cong^B_A$ is a DURG structure on a type family $B : A \to \mathrm{Type}$, and $f : C \to A$ is any map, then we can *pull back* the displayed relation to give a family of reflexive graphs $\langle B\ (f\ c), \cong^B_{\rho(f\ c)}\rangle$ indexed by $c : C$.

**Theorem 2.11.** *Let* $f : A \simeq A'$ *be an equivalence of types which carry URG structures. Furthermore, let* $\cong^B_A$ *and* $\cong^{B'}_{A'}$ *be DURG structures. Any fiberwise relational isomorphism* $g$ *between the underlying family of reflexive graphs of* $\cong^B_A$ *and the pullback of* $\cong^{B'}_{A'}$ *along* $f$*, over* $A$*, induces an equivalence of total spaces*

$$(a : A) \times B\ a \simeq (a' : A') \times B'\ a'.$$

*Proof.* As for (non-displayed) univalent graphs, we obtain a fiberwise equivalence $(a : A) \to B\ a \simeq B'(f\ a)$ from $g$. According to [17, Thm. 4.7.7], this yields an equivalence

$$(a : A) \times B\ a \simeq (a : A) \times B'(f\ a).$$

A pseudo-inverse of $f$ induces a pseudo-inverse of

$$\lambda\,\langle a, b\rangle \;\mapsto \langle f\ a, b\rangle : (a : A) \times B'(f\ a) \to (a' : A') \times B'\ a'. \qquad \square$$

Note the asymmetry of the previous statement. An $a : A$ is fixed and we prove an isomorphism between the structure that $B$ adds to $a$, and that which $B'$ adds to $f\ a$, the element $a$ viewed as a term of type $A'$. A more symmetric version could be obtained by pulling back along $f$ to prove that the fiberwise maps cancel on the left, and along $f^{-1}$ to prove that they cancel on the right.

Let us discuss alternative definitions of (D)URGs. Instead of the second and third axioms in the definition of a (D)URG one could pack the same data in the family of equivalences

$$\overline{\mathrm{uni}} : (a, a' : A) \to (a \cong_A a') \simeq (a =_A a'),$$

and

$$\overline{\overline{\mathrm{uni}}} : \{a : A\} \to (b, b' : B\ a) \to (b \cong^B_{\rho\, a} b') \simeq (b =_{B\ a} b'),$$

respectively. However, being able to choose the reflexivity term turns out to be more practical than having to deal with the inverse image $\rho\, a$ of refl under the univalence equivalence.

For the cubical setting, Evan Cavallo suggested the following notion of displayed univalent graph:

$$\begin{aligned}\overline{\overline{\overline{\mathrm{uni}}}} : \{a, a' : A\} \to (b : B\ a) \to (p : a \cong_A a') \to (b' : B\ a')\\ \to (b \cong^B_p b') \simeq \mathrm{PathP}\ (\lambda\, i \;\mapsto B(\overline{\mathrm{uni}}\ a\ a'\ p\ i))\ b\, b'\end{aligned}$$

Why are $\overline{\overline{\overline{\mathrm{uni}}}}$ and the displayed $\overline{\overline{\mathrm{uni}}}$ equivalent choices? The underlying URG satisfies identity induction, but for the relation rather than paths. This "relational J" can be used to reduce $\overline{\overline{\overline{\mathrm{uni}}}}$ to $\overline{\overline{\mathrm{uni}}}$.

The formulation with $\overline{\overline{\overline{\mathrm{uni}}}}$ is advantageous in cubical type theory, because there we have other ways to establish equivalences besides path induction, so the reflexivity terms play a less crucial role. We have formalized both versions, but chose the version with $\overline{\overline{\overline{\mathrm{uni}}}}$ to merge into the *cubical* library. However, for standard HoTT or informal univalent mathematics, these dependent paths are less practical. Therefore, we have presented the framework using the reflexivity formulation here.

Regardless of the particular choice of type theory and (D)URG structure, a (displayed) univalent graph structure is reflexive in any case. So, in a general context, these structures may simply be called "(displayed) univalent graph structures", omitting the "reflexive" part when it is not an explicit axiom.

# 3 Equivalence of Strict 2-groups and Crossed Modules

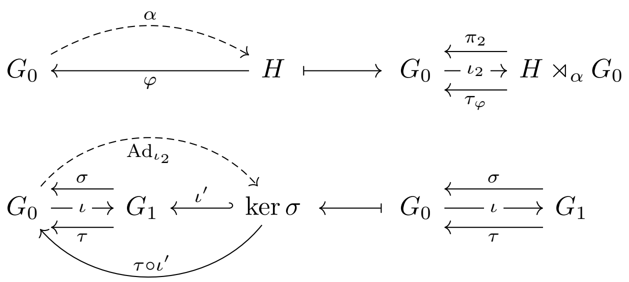
A strict 2-group is an internal category

$$G_1 \times_{G_0} G_1 \xrightarrow{\;\circ\;} G_1 \underset{\tau}{\overset{\sigma}{\underset{\longleftarrow \iota \longrightarrow}{\longrightarrow}}} G_0$$

in the category of groups. The group homomorphisms $\sigma, \tau, \iota$ and $\circ$ are the source, target, identity-assigning and composition operations. They are subject to the following coherence conditions. The composition $\circ$ is associative and satisfies left and right unit laws, and the source and target of identity and composite morphisms behave as expected. The homomorphism property of $\circ$ is known as the interchange law. It states that $(c \cdot_1 d) \circ (a \cdot_1 b) = (c \circ a) \cdot_1 (d \circ b)$.

A crossed module $G_0 \overset{\alpha}{\underset{\varphi}{\longleftarrow}} H$ consists of a group homomorphism $\varphi$ and an action of $G_0$ on the group $H$, $\alpha : G_0 \times H \to H$. We write the action infix. Moreover, $\varphi$ needs to be $\alpha$-equivariant, that is $\varphi(g \, \alpha \, h) = g(\varphi \, h)g^{-1}$, and $\alpha$ has to satisfy the so-called Peiffer-identity $(\varphi \, h) \, \alpha \, h' = hh'h^{-1}$.

Strict 2-groups and crossed modules are equivalent as categories. In the following we use DURG structures to establish their type-theoretical incarnation as a tower of structure over a group and prove that the two maps




define an equivalence of types. Here Ad denotes the adjoint action.

We begin with the equivalence of split monomorphisms and group actions.

From the SIP for groups we obtain a URG structure on the type of groups. As described in Lemma 2.6, this also induces a URG structure $\cong_{\mathrm{Grp}^2}$ on the type of pairs of groups.

**Proposition 3.1.** *Homomorphisms can be displayed over pairs of groups. In other words, there is a DURG structure on the type family*

$$\lambda \, \langle G, H \rangle \mapsto (G \to_{\mathrm{Grp}} H).$$

*Proof.* An element of $\langle G, G' \rangle \cong_{\mathrm{Grp}^2} \langle H, H' \rangle$ is a pair $\langle p, q \rangle$ of isomorphisms. The displayed relation is given by commutativity of the following square:

$$\begin{array}{ccc} G & \xrightarrow{f} & H \\ {\scriptstyle p}\downarrow{\scriptstyle \wr} & & {\scriptstyle \wr}\downarrow{\scriptstyle q} \\ G' & \xrightarrow[f']{} & H' \end{array}$$

Reflexivity of this relation with respect to the identity group isomorphisms on $G$ and $H$ is trivial. The relational singleton at $f$ is

$$(f' : G \to_{\mathrm{Grp}} H) \times f \cong_{\langle \mathrm{id}, \mathrm{id} \rangle} f'.$$

Since function extensionality extends to group homomorphisms, the second component is equivalent to $f = f'$, and the whole $\Sigma$-type to the $=$-singleton at $f$. $\square$

In a similar fashion we can define a DURG structure on $\lambda \langle G, H\rangle \mapsto (H \to_{\mathrm{Grp}} G)$. These two can be combined to obtain a DURG structure on $\lambda \langle G, H\rangle \mapsto (G \to_{\mathrm{Grp}} H) \times (H \to_{\mathrm{Grp}} G)$. Taking the total space we obtain a URG structure on the type of pairs of groups with homomorphisms back and forth. Being a section-retraction-pair of group homomorphisms is a universally quantified equality statement in a set and thus a mere proposition. Hence, Proposition 2.4 can be used to display this split condition.

**Proposition 3.2.** *Group actions can be displayed over pairs of groups.*

*Proof.* As above we define the displayed relation for group action structures to be commutativity of this square:

$$\begin{array}{ccc} G \times H & \xrightarrow{\alpha} & H \\ {\scriptstyle \langle p,q\rangle}\big\downarrow\wr & & \wr\big\downarrow{\scriptstyle q} \\ G' \times H' & \xrightarrow[\beta]{} & H' \end{array}$$

Since the axioms for group actions are mere propositions, they can easily be imposed using Proposition 2.4. □

**Theorem 3.3.** *The type of split monomorphisms in groups is equivalent to the type of actions.*

*Proof.* By multiple applications of Theorem 2.8 we obtain DURG structures on the type family that assigns to a group $G$ the type of split monomorphisms with source $G$, and also the type family that assigns to $G$ the type of actions of $G$ on another group. We want to apply Theorem 2.11 to the identity equivalence on the type of groups.

Fix a group $G_0$. It is easy to verify that for an action $G_0 \overset{\alpha}{\dashrightarrow} H$, there is a split monomorphism $G_0 \underset{\iota_2}{\overset{\pi_2}{\leftrightarrows}} H \rtimes_\alpha G_0$. Conversely, given a split mono $G_0 \underset{\iota}{\overset{\sigma}{\leftrightarrows}} G_1$, the adjoint action of $G_1$ on $\ker \sigma$ can be extended to $G_0$ by precomposing with $\iota$.

Having constructed the maps back and forth, it remains to be shown that they are relationally inverse to each other.

On the left, this requires an isomorphism $\varphi$ such that

$$\begin{array}{ccc} G_0 \times \ker \pi_2 & \xrightarrow{\mathrm{Ad}_{\iota_2}} & \ker \pi_2 \\ {\scriptstyle \langle \mathrm{id},\varphi\rangle}\big\downarrow & & \big\downarrow{\scriptstyle \varphi} \\ G_0 \times H & \xrightarrow[\alpha]{} & H \end{array}$$

commutes. An element of $\ker \pi_2$ consists of an $h : H$, $g : G_0$ and a proof that $g = 1_{G_0}$. We leave it to the reader to verify that the $\varphi$ that extracts the $H$-component satisfies above requirements.

On the right, we need to construct an isomorphism $\psi$ such that the squares

$$\begin{array}{ccc} \ker \sigma \rtimes_{\mathrm{Ad}_\iota} G_0 & \xrightarrow{\pi_2} & G_0 \\ {\scriptstyle \psi}\big\downarrow & & \big\downarrow{\scriptstyle \mathrm{id}} \\ G_1 & \xrightarrow[\sigma]{} & G_0 \end{array} \quad \text{and} \quad \begin{array}{ccc} G_0 & \xrightarrow{\iota_2} & \ker \sigma \rtimes_{\mathrm{Ad}_\iota} G_0 \\ {\scriptstyle \mathrm{id}}\big\downarrow & & \big\downarrow{\scriptstyle \psi} \\ G_0 & \xrightarrow[\iota]{} & G_1 \end{array}$$

commute. A valid choice is $\psi\langle h, g, p\rangle :\equiv h\,\iota\, g$. □

In the following we show the equivalence of precrossed modules and internal reflexive graphs (in the category of groups) by building on the equivalence from Theorem 3.3.

**Definition 3.4.** A *precrossed module* consists of an action of $G_0$ on $H$ together with an equivariant homomorphism $\varphi : H \to_{\mathrm{Grp}} G_0$. An *internal reflexive graph* consists of a split monomorphism $G_0 \underset{\iota}{\overset{\sigma}{\leftrightarrows}} G_1$ with a second retraction $\tau : G_1 \to_{\mathrm{Grp}} G_0$.

We may display the precrossed module structure over actions by lifting the homomorphism structure twice using Lemma 2.7, taking the total space, and imposing equivariance using Proposition 2.4. As always, Theorem 2.8 reassociates these layers of structure. A similar argument allows us to display the second retraction over split monomorphisms.

**Theorem 3.5.** *The type of precrossed modules is equivalent to the type of internal reflexive graphs.*

*Proof.* We aim to invoke Theorem 2.11. Fix an action $\alpha$ of $G_0$ on $H$ and its corresponding split mono $G_0 \underset{\iota_2}{\overset{\pi_2}{\leftrightarrows}} H \rtimes_\alpha G_0$. We need to construct relationally inverse maps that turn an $\alpha$-equivariant homomorphism $\varphi$ into a retraction $\tau_\varphi$ of $\iota_2$, and any retraction $\tau$ of $\iota_2$ into an $\alpha$-equivariant homomorphism $\varphi_\tau$. For the first direction, we define $\tau_\varphi\langle h, g\rangle :\equiv (\varphi\, h)g$. Conversely, we put $\varphi_\tau :\equiv \tau \circ_{\mathrm{Grp}} \iota_1$ with $\iota_1 : \ker \pi_1 \hookrightarrow_{\mathrm{Grp}} H \rtimes_\alpha G_0$ the first inclusion and $\pi_1$ the first projection of the semidirect product. To prove that these constructions are relationally inverse is to give homotopies $\tau_\varphi \circ \iota_1 \sim \varphi$ and $\tau \sim \tau_\varphi$. The details of these computations can be found in the formalization. □

The Peiffer condition for internal reflexive graphs is not well known. It is essentially the Peiffer condition for crossed modules under the transformation from Theorem 3.5.

**Definition 3.6.** The *Peiffer condition* for an internal reflexive graph $G_0 \;\overset{\sigma}{\underset{\tau}{\leftleftarrows\!\!\!\!\to}}\; G_1$ is

$$(a, b : G_1) \to (\iota(\sigma\, b))a(\iota(\sigma\, a^{-1}))(\iota(\sigma\, b^{-1}))b(\iota(\tau\, a)) = ba.$$

**Theorem 3.7.** *The type of crossed modules is equivalent to the type of Peiffer graphs.*

*Proof.* Being mere propositions, it is easy to display the Peiffer conditions for precrossed modules and internal reflexive graphs. Moreover, the statement of the theorem follows from the logical equivalence of the Peiffer condition for a precrossed module and the Peiffer condition of its corresponding internal reflexive graph as constructed in Theorem 3.5. The calculations to verify this logical equivalence can be found in the formalization. □

A strict 2-group is an internal reflexive graph together with a vertical composition operation. We prove that Peiffer graphs are equivalent to strict 2-groups.

**Definition 3.8.** Fix an internal reflexive graph $\mathcal{G}$ as in Definition 3.6. Two arrows $b, a : G_1$ are *composable*, if $\sigma\, b = \tau\, a$. A *vertical composition operation* on that graph consists of a map

$$\_ \circ_{\_} \_ : (b, a : G_1) \to \text{isComposable}\, b\; a \to G_1$$

together with proofs that $\circ$ is respected by $\tau$ and $\sigma$, that it is a homomorphism, associative, and satisfies left and right unit laws with respect to $\iota$.

**Lemma 3.9.** *If $\circ$ is a vertical composition operation, then $b \circ_p a = b(\iota(\sigma\, b^{-1}))a$.*

The classical proof of Lemma 3.9 can be found in [16]. It goes through in type theory.

**Corollary 3.10.** *The type of vertical compositions on $\mathcal{G}$ is a mere proposition.*

**Theorem 3.11.** *The type of Peiffer graphs is equivalent to the type of internal reflexive graphs with a vertical composition operation.*

*Proof.* We use Proposition 2.4 to display the vertical compositions over internal reflexive graphs. Using Theorem 2.11 once again, it suffices to prove that the Peiffer condition is logically equivalent to the type of vertical compositions on $\mathcal{G}$.

If $\mathcal{G}$ is Peiffer, we need to show that $b \circ_p a :\equiv b(\iota(\sigma\, b^{-1}))a$ has the properties of a vertical composition. The verification of the homomorphism property is not too difficult. The other properties are satisfied by $\circ$ even if $\mathcal{G}$ is not Peiffer. $\square$

# 4 Higher Groups

This section is concerned with higher groups as introduced in [9]. After establishing some results about pointed types we prove the SIP for $(n, k)$-groups and their homomorphisms. This is succeeded by the definition of the first Eilenberg–MacLane space in cubical type theory and a proof that $(0, 1)$-groups are equivalent to axiomatic groups.

A universe is univalent if and only if $A \cong B :\equiv A \simeq B$ defines a URG structure. We may display pointedness over the universe by showing that $a \cong_e b :\equiv e\; a = b$ defines a DURG structure on $\lambda\,(A : \text{Type}) \mapsto A$. Note that this is beyond the scope of displayed categories, since there's no truncation requirement. Taking the total space of this DURG structure shows that pointed equivalences characterize the identity types of pointed types.

Let $\langle A, *_A \rangle$ be a pointed type. A *pointed family* is a type family $B : A \to \text{Type}$ together with a base point $*_B : B\; *_A$. A pointed section consists of a function $f : (a : A) \to B\; a$ and a proof $f_* : f\; *_A = *_B$. For two pointed sections we have the two kinds of pointed homotopies

$$f \sim_* g :\equiv (a : A) \to_* \langle f\; a = g\; a, f_* \cdot {g_*}^{-1} \rangle$$

and

$$f \sim_*^{\mathrm{P}} g :\equiv (H : f \sim g) \times \text{PathP}\,(\lambda\, i \mapsto H\; *_A\; i = *_B)\; f_*\; g_*.$$

**Theorem 4.1.** *Pointed function extensionality holds, i.e., any two pointed sections $f$ and $g$ satisfy*

$$(f \sim_* g) \simeq (f \sim_*^{\mathrm{P}} g) \simeq (f = g). \tag{1}$$

*Proof.* The two kinds of pointed homotopies consist of an $H : f \sim g$ and a filler of these squares:

$$
\begin{array}{ccc}
f \ast_A & \xrightarrow{f_\ast \cdot {g_\ast}^{-1}} & g \ast_A \\
\| & & \| \\
f \ast_A & \xrightarrow[H \ast_A]{} & g \ast_A
\end{array}
\qquad
\begin{array}{ccc}
g \ast_A & \xrightarrow{g_\ast} & \ast_B \\
{\scriptstyle H \ast_A}\uparrow & & \| \\
f \ast_A & \xrightarrow[f_\ast]{} & \ast_B
\end{array}
$$

The cubical groupid laws and univalence can be used to show that

$$
\begin{aligned}
(H\ast_A = f_\ast \cdot {g_\ast}^{-1}) &= ((H\ast_A)^{-1} \cdot f_\ast = (H\ast_A)^{-1} \cdot (H\ast_A \cdot g_\ast)) \\
&= ((H\ast_A)^{-1} \cdot f_\ast \cdot \text{refl} = g_\ast) \\
&= \text{PathP } (\lambda\, i \mapsto H \ast_A i = \ast_B)\ f_\ast\ g_\ast.
\end{aligned}
$$

Transport along this path induces the first equivalence in (1). The second equivalence is given by the the map that sends $\langle H, H_\ast \rangle$ to the path $\lambda\, i \mapsto \langle \lambda\, a \mapsto H\ a\ i, H_\ast\ i \rangle$. ☐

**Definition 4.2.** For integers $n \geq 0$ and $k \geq 1$, we define the type

$$\langle n, k \rangle\text{Grp} :\equiv (B^k G : \text{Type}) \times B^k G \times \text{isConn}_{k-1}\, B^k G \times \text{isTrunc}_{n+k}\, B^k G$$

of *$k$-tuply groupal $n$-groupoids* or *$(n, k)$-groups*.[4]

Connectedness and truncatedness are mere propositions. By displaying these properties over pointed types and taking the total space we obtain a URG structure on $\langle n, k \rangle$Grp, and we see that pointed equivalences characterize the identity types of $(n, k)$-groups.

**Proposition 4.3.** *Homomorphisms of $(n, k)$-groups can be displayed over pairs of $(n, k)$-groups. The displayed relation on the family $\lambda \langle B^k G, B^k H \rangle \mapsto (B^k G \to_\ast B^k H)$ is given by the type of pointed homotopies filling the following square:*

$$
\begin{array}{ccc}
B^k G & \xrightarrow[\ast]{f} & B^k H \\
{\scriptstyle p}\downarrow_\ast \wr & & {\scriptstyle q}\downarrow_\ast \wr \\
B^k G' & \xrightarrow[\ast]{f'} & B^k H'
\end{array}
$$

This uses pointed function extensionality. As a consequence, we see that the identity types of $\langle n, k \rangle$Grp-homomorphisms are just homotopies of the underlying functions. This is used in the proof of the next theorem (cf. [9, p. 9]).

**Theorem 4.4.** *The type $B^k G \to_\ast B^k H$ is $n$-truncated and $\langle n, k \rangle$Grp is $(n+1)$-truncated.*

In the following we construct the first Eilenberg–MacLane space of a group using higher inductive type in the setting of cubical type theory, and we show that this construction is a right inverse to the first homotopy group. The main reference for the construction in standard HoTT is [14].

[4] A slightly better name is perhaps *$k$-symmetric $(n + 1)$-groups*. Here we stick to the terminology of [9].

**Definition 4.5.** The *first Eilenberg–MacLane space* of a group $G$ is the higher inductive type $\mathcal{E}_1\, G$, abbreviated to $\mathcal{E}$, with constuctors

$$\begin{aligned} \text{base}_{\mathcal{E}} &: \mathcal{E}; \\ \text{loop}_{\mathcal{E}} &: G \to \text{base}_{\mathcal{E}} = \text{base}_{\mathcal{E}}; \\ \text{comp}_{\mathcal{E}} &: (g\ h : G) \to \text{PathP}\ (\lambda\, i\ \mapsto \text{base}_{\mathcal{E}} = \text{loop}_{\mathcal{E}} h\ i)\ (\text{loop}_{\mathcal{E}} g)\ (\text{loop}_{\mathcal{E}}(gh)); \\ \text{squash}_{\mathcal{E}} &: \text{isTrunc}_1\, \mathcal{E}\,. \end{aligned}$$

The $\text{comp}_{\mathcal{E}}$ constructor fills the square

$$\begin{array}{ccc} a & \xrightarrow{\text{loop}_{\mathcal{E}}(gh)} & c \\ \| & & \uparrow{\scriptstyle \text{loop}_{\mathcal{E}} h} \\ a & \xrightarrow[\text{loop}_{\mathcal{E}} g]{} & b. \end{array}$$

An equivalent condition is $\text{loop}_{\mathcal{E}}(gh) = \text{loop}_{\mathcal{E}} g\ \cdot\ \text{loop}_{\mathcal{E}} h$, but the square is more convenient to reason about in cubical type theory.

**Theorem 4.6.** *Let $B : \mathcal{E} \to \text{Type}^{\leq 1}$ be a family of groupoids over $\mathcal{E}$, and $* : B\ \text{base}_{\mathcal{E}}$. If there is a map*

$$\text{toLoop} : (g : G) \to \text{PathP}\ (\lambda\, i\ \mapsto B\ (\text{loop}_{\mathcal{E}} g\ i))\ *\ *$$

*such that the dependent square*

$$\begin{array}{ccc} * & \xrightarrow{\text{toLoop}(gh)} & * \\ \| & & \uparrow{\scriptstyle \text{toLoop}\, h} \\ * & \xrightarrow[\text{toLoop}\, g]{} & * \end{array} \tag{2}$$

*has a filler* toIsComp *for all $g\ h : G$, then there is a function $f : (x : \mathcal{E}) \to_* B\ x$.*

*Proof.* We define $f$ by $\mathcal{E}$-induction:

$$\begin{aligned} f\ \text{base}_{\mathcal{E}} &:\equiv *_B \\ f(\text{loop}_{\mathcal{E}}\ g\ i) &:\equiv \text{toLoop}\ \ g\ i \\ f(\text{comp}_{\mathcal{E}}\ g\ h\ i\ j) &:\equiv \text{toIsComp}\ \ g\ h\ i\ j \end{aligned}$$

The fourth case, a path over $\text{squash}_{\mathcal{E}}$, asks for a proof that $B$ is a 1-type over $\mathcal{E}$, but that follows from the assumptions. □

Elimination into the proposition $\text{isConn}_0\, \mathcal{E}$ shows that $\mathcal{E}$ is connected. In particular, it is a $(0,1)$-group. If $B$ is constant, Theorem 4.6 becomes the recursion principle

$$(G \to_{\text{Grp}} \pi_1 \langle B, *_B \rangle) \to (\mathcal{E}_1\, G \to_* \langle B, *_B \rangle).$$

**Theorem 4.7.** *There is an equivalence*

$$\langle 0,1 \rangle \text{Grp} \underset{\mathcal{E}}{\overset{\pi_1}{\rightleftarrows}} \text{Grp}.$$

The rest of this section constitutes a proof of this theorem. We have established URG structures on both $\langle 0, 1\rangle\mathrm{Grp}$ and Grp, so it suffices to show that the two maps are relational inverses. We use parts of the adjunction

$$(\mathcal{E}_1\, H \to_* BG) \simeq (H \to_{\mathrm{Grp}} \pi_1\, BG).$$

**Theorem 4.8.** *The group $\pi_1(\mathcal{E}_1\, G)$ is isomorphic to $G$.*

This is proved using the encode-decode method. It shows cancellation on the right in Theorem 4.7.

**Proposition 4.9.** *There is a map*

$$\varphi : (\pi_1(\mathcal{E}_1\, H) \to_{\mathrm{Grp}} \pi_1\, BG) \to (\mathcal{E}_1\, H \to_* BG)$$

*which restricts to isomorphisms.*

*Proof.* Let $f : \pi_1(\mathcal{E}_1\, H) \to_{\mathrm{Grp}} \pi_1\, BG$. From Theorem 4.8 we have a $g : H \to_{\mathrm{Grp}} \pi_1(\mathcal{E}_1\, H)$. Put

$$h :\equiv f \circ_{\mathrm{Grp}} g : H \to_{\mathrm{Grp}} \pi_1\, BG.$$

The type $BG$ is a 1-type, so by $\mathcal{E}_1\, H$-recursion we have a map $h' : \mathcal{E}_1\, H \to BG$. Pointedness of $h'$ is trivial. Being a map between pointed connected types, $h'$ is surjective.

Assume now that $f$ is an isomorphism to begin with. If we can show that $h'$ is an embedding, then it is also an equivalence (cf. [17, Theorem 4.6.3]).

By the elimination principle for pointed connected types, it suffices to show that $\mathrm{ap}_{h'} : (x = y) \to (h'\, x = h'\, y)$ is an equivalence at $x \equiv y \equiv \mathrm{base}_{\mathcal{E}}$. Let $e$ be the equivalence from Theorem 4.8. There is a trivial homotopy $f \circ e \sim \mathrm{ap}_{h'}$. The 2-out-of-3 property of equivalences implies that $\mathrm{ap}_{h'}$ is an equivalence. □

We show cancellation on the left in Theorem 4.7. Let $BG : \langle 0, 1\rangle\mathrm{Grp}$ and $f : \pi_1(\mathcal{E}_1(\pi_1\, BG)) \to_{\mathrm{Grp}} \pi_1\, BG$ be the isomorphism from Theorem 4.8. Applying Proposition 4.9 to $H :\equiv \pi_1\, BG$ and $f$ produces $\mathcal{E}_1(\pi_1\, BG) \simeq_* BG$.

# 5 Conclusion and Future Work

For many kinds of composite structure, the natural notion of equivalence is built up from the notions of equivalence of the component parts in a canonical way, using the constructions implemented in Section 2. A natural next step is to use reflection to automate this process. This has been achieved for the special case of standard notion of structure in [6], and we are working on extending that work to cover (D)URGs in general.

An intriguing variation of this would be to use that reflexive graphs form a presheaf $(\infty, 1)$-topos, with the URGs as a subtopos. These thus carry the structures of a model of type theory. Indeed, DURGs correspond to type families in the model in URGs. If we could internalize these models, this would provide an alternate way of automating the construction of URG structures.

Another natural direction will be to facilitate the extension of a URG structure to a compatible category or bicategory structure, in which the isomorphisms become the univalent relation. Ideally, this will also be automated, so that for instance 1-types of algebraic structures can automatically be given a compatible univalent category structure and underlying URG structure.

In Section 4 we constructed the delooping of a group as a higher inductive type. A natural next step will be to define the type of coherent weak 2-groups and prove this is equivalent to the type of 2-groups. This will similarly involve defining the delooping of a weak 2-group as a higher inductive type. Finally, we can then hope to prove that the identifications between 2-groups presented by crossed modules are given by invertible butterflies [5], and that every 2-group can be presented by a crossed module if the principle that *sets cover 1-types* is assumed. (This says that for every 1-type $X$ there merely exists a set $Y$ and a surjection $f : Y \to X$.)

## References


[1] Peter Aczel. Homotopy type theory and the structure identity principle, 2012. Talk notes. URL: https://www.newton.ac.uk/files/seminar/20120207160016301-153011.pdf.

[2] Benedikt Ahrens, Dan Frumin, Marco Maggesi, and Niels van der Weide. Bicategories in Univalent Foundations. In Herman Geuvers, editor, *4th International Conference on Formal Structures for Computation and Deduction (FSCD 2019)*, volume 131 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 5:1–5:17, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.FSCD.2019.5.

[3] Benedikt Ahrens and Peter LeFanu Lumsdaine. Displayed categories. *Log. Methods Comput. Sci.*, 15(1):Paper No. 20, 18, 2019. doi:10.23638/LMCS-15(1:20)2019.

[4] Benedikt Ahrens, Paige Randall North, Michael Shulman, and Dimitris Tsementzis. The univalence principle, 2021. arXiv:2102.06275.

[5] Ettore Aldrovandi and Behrang Noohi. Butterflies. I. Morphisms of 2-group stacks. *Adv. Math.*, 221(3):687–773, 2009. doi:10.1016/j.aim.2008.12.014.

[6] Carlo Angiuli, Evan Cavallo, Anders Mörtberg, and Max Zeuner. Internalizing representation independence with univalence. *Proc. ACM Program. Lang.*, 5(POPL), January 2021. arXiv:2009.05547, doi:10.1145/3434293.

[7] Danil Annenkov, Paolo Capriotti, Nicolai Kraus, and Christian Sattler. Two-level type theory and applications, 2019. arXiv:1705.03307.

[8] Ronald Brown and Christopher B. Spencer. Double groupoids and crossed modules. *Cahiers Topologie Géom. Différentielle*, 17(4):343–362, 1976. URL: www.numdam.org/item/CTGDC_1976__17_4_343_0/.

[9] Ulrik Buchholtz, Floris van Doorn, and Egbert Rijke. Higher groups in homotopy type theory. In *Proc. 33rd ACM/IEEE Symp. Logic Comp. Sci.*, LICS '18, page 205–214, 2018. arXiv:1802.04315, doi:10.1145/3209108.3209150.

[10] Cyril Cohen, Thierry Coquand, Simon Huber, and Anders Mörtberg. Cubical Type Theory: A Constructive Interpretation of the Univalence Axiom. In Tarmo Uustalu, editor, *21st International Conference on Types for Proofs and Programs (TYPES 2015)*, volume 69 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 5:1–5:34, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.TYPES.2015.5.

[11] Thierry Coquand and Nils Anders Danielsson. Isomorphism is equality. *Indagationes Mathematicae*, 24(4):1105 – 1120, 2013. In memory of N.G. (Dick) de Bruijn (1918–2012). doi:https://doi.org/10.1016/j.indag.2013.09.002.

[12] Thierry Coquand, Simon Huber, and Anders Mörtberg. On higher inductive types in cubical type theory. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '18, page 255–264, New York, NY, USA, 2018. Association for Computing Machinery. doi:10.1145/3209108.3209197.

[13] Martín Hötzel Escardó. Introduction to Univalent Foundations of Mathematics with Agda, 2020. URL: https://www.cs.bham.ac.uk/~mhe/HoTT-UF-in-Agda-Lecture-Notes/HoTT-UF-Agda.html, arXiv:1911.00580.

[14] 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.

[15] Anders Mörtberg and Andrea Vezzosi. An experimental library for cubical agda. URL: https://github.com/agda/cubical.

[16] Sven-S. Porst. Strict 2-groups are crossed modules, 2008. arXiv:0812.1464.

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

[18] Andrea Vezzosi, Anders Mörtberg, and Andreas Abel. Cubical agda: A dependently typed programming language with univalence and higher inductive types. *Proc. ACM Program. Lang.*, 3(ICFP), July 2019. doi:10.1145/3341691.

[19] Jakob von Raumer. Formalization of non-abelian topology for homotopy type theory, 5 2015. URL: http://von-raumer.de/academic/msc-thesis.pdf.