HIGHER SCHREIER THEORY IN CUBICAL AGDA

DAVID JAZ MYERS AND ZYAD YASSER

Abstract. Homotopy type theory (HoTT) enables reasoning about groups directly as the types of symmetries (automorphisms) of mathematical structures. The HoTT approach to groups—first put forward by Buchholtz, van Doorn, and Rijke—identifies a group with the type of objects of which it is the symmetries. This type is called the "delooping" of the group, taking a term from algebraic topology. This approach naturally extends the group theory to higher groups which have symmetries between symmetries, and so on. In this paper, we formulate and prove a higher version of Schreier's classification of all group extensions of a given group. Specifically, we prove that extensions of a group G by a group K are classified by actions of G on a delooping of K. Our proof is formalized in Cubical Agda, a dependently typed programming language and proof assistant which implements HoTT.

§1. Introduction.

An extension of a group G by a group K is a short exact sequence

$$0 \to K \xrightarrow{i} E \xrightarrow{p} G \to 0.$$

That is, a sequence of homomorphisms in which i is injective, p is surjective, and $\operatorname{im} i = \ker p$. As a set, any extension E is bijective with the cartesian product $K \times G$; but there may be many different group structures on this cartesian product which are, in some sense, built out of the group structures of G and K. Two extensions E and E' are isomorphic when there is an isomorphism $E \cong E'$ commuting with the inclusions i and projections p; we define $\mathrm{Ext}(G, K)$ to be set of isomorphism classes of extensions of G by K.

In 1926 [27], Otto Schreier gave a series of cocycle conditions which classified all the possible extensions E of G by K. In doing so, he inaugurated the field of *Schreier theory*: the study of extensions of algebraic structures, and in particular of group-like structures. In 1934, Baer [6] examined the abelian extensions of abelian groups in terms of presentations of G by generators and relations, and discovered an abelian group law on $\mathrm{Ext}(G, K)$. While working on computing Baer groups, Eilenberg, and Mac Lane noticed a similarity between those computations and the ones done in the cohomology of spaces. In a series of three papers [13–15], they developed a cohomology theory for groups and showed that $H^2(G; K)$ classified *central* extensions of G by an abelian group K—those extensions for which K lies in the center ZG of G. They also related group cohomology to the cohomology of spaces by developing the *Eilenberg–Mac Lane* spaces $\mathsf{K}(G, n)$, and showing that

Received June 9, 2024.

2020 *Mathematics Subject Classification.* Primary 03B38, Secondary 55P47.

Key words and phrases. homotopy type theory, Schreier theory, higher groups.

The authors appreciate the support of Tamkeen under the NYU Abu Dhabi Research Institute Grant CG008.

0022-4812/00/0000-0000
DOI:10.1017/jsl.2024.78

$H^2(G;K)$, and hence central extensions, were in bijection with homotopy classes of pointed maps $\mathsf{K}(G,1) \to_* \mathsf{K}(K,2)$.

This leads naturally to the following question: is there an object associated with a group K which classifies *all* extensions by a possibly non-abelian K? Building on work on *non-abelian cohomology* by Dedecker and Luks [11, 12], Grothendieck answered this question affirmatively with his Grothendieck construction [17], showing that there is a *2-group* $\mathsf{BAut}(\mathsf{B}K)$ so that pseudo-functors from a delooping $\mathsf{B}G$ of G to $\mathsf{BAut}(\mathsf{B}K)$ correspond to extensions of G by K. This result was extended to stacks by Giraud [16] and to stacks of 2-groups by Breen [8].

In this paper, we will extend this classification of extensions to all (stacks of) *higher groups* [9] with an elementary and surprisingly concrete proof in homotopy type theory (HoTT).

THEOREM 1.1 (Higher Schreier theorem). *For any higher groups G and K, the type $\mathrm{EXT}(G;K)$ of extensions of G by K is equivalent to the type of pointed functions $\mathsf{B}G \to_* \mathsf{BAut}(\mathsf{B}K)$, or, in other words, the type of actions of G on a delooping $\mathsf{B}K$ of K.*

This result was first proved informally in [26]; here, we present a formal proof of the higher Schreier theorem in Cubical Agda [31]. It straightforwardly generalizes the elementary classification of split extensions by homomorphic actions of G on K familiar from any undergraduate textbook on group theory.

While Schreier's original classification was concrete in the sense that it gave explicit cocycle conditions determining the group law of the extension, our construction is concrete in that it directly defines a type of mathematical structures whose group of symmetries gives the extension. We are able to do this by working in HoTT, where any object comes naturally equipped with a notion of *identification*—equality, isomorphism, structure-preserving equivalence, etc.—so that groups may be viewed as the self-identifications or symmetries of a object natively. Rather than expressing the group law of an extension E decomposes into the group laws of G and K via cocycles, Theorem 1.1 shows how the structures whose symmetries form the group E decompose into structures that whose symmetries form the group G and K, respectively.

As an example of this way of thinking, consider the short exact sequence

$$0 \to \mathbb{R}^4 \to \text{Poincaré} \to \text{Lorentz} \to 0$$

witnessing the Poincaré group as an extension of the Lorentz group by the group of translational symmetries of Minskowski space. We will see this series of group homomorphisms as the action on symmetries of a *fiber sequence* of types:

$$\text{Affine}(\mathbb{R}^4) \to \text{MinkowskiSpace} \to \text{LorentzSpace}$$

where $\text{Affine}(\mathbb{R}^4)$ is the type of 4-dimensional real affine spaces, LorentzSpace is the type of Lorentz spaces (an 4-dimensional real vector space with an inner product of signature $3+1$), and MinkowskiSpace is the type of Minkowski spaces (an affine space over a Lorentz space, as defined for example in [29, p. 231]); the first function considers the real affine space as an affine space over the canonical Lorentz space $\mathbb{R}^{3+1}$, and the second projects out the underlying Lorentz space. The corresponding classifying map $\text{LorentzSpace} \to_* \mathsf{BAut}(\text{Affine}(\mathbb{R}^4))$ determined by

https://doi.org/10.1017/jsl.2024.78 Published online by Cambridge University Press

the higher Schreier theorem sends a Lorentz space V to the type $\mathrm{Affine}(V)$ of affine spaces over V; this evidently sends $\mathbb{R}^{3+1}$ to $\mathrm{Affine}(\mathbb{R}^4)$.

In Section 2, we give a quick introduction to Cubical Agda, the dependently typed programming language in which our results are formalized. After that, in Section 3 we recall the theory of higher groups in HoTT. With this under our belt, we make short work of our main theorem in Section 4; in Section 6 we show how it generalizes the usual classification of split extensions by homomorphic actions.

§2. A brief introduction to homotopy type theory and Cubical Agda. HoTT [30] is a novel foundation of mathematics based on Martin–Löf dependent type theory (MLTT) [23] with inspiration from modern homotopy theory. Through its basis in intuitionistic type theory, HoTT gives a computational perspective on mathematical proofs: proofs may be seen as programs that compute witnesses to the truth of a proposition. HoTT emerged from the work of Awodey and Warren [5] and Voevodsky [32] on the homotopical interpretation of MLTT, extending Hoffman and Streicher's groupoid model of MLTT [18].

In type theory, there are four primitive *judgements*: that something is a *type* (denoted $A : \mathsf{Type}$), that something is an *element* of a type (denoted $a : A$), that two types are equal by definition (denoted $A := B$) and that two elements are equal by definition (denoted $a := b$). Formally, type theory is a system for moving from some constellation of these judgements to others via given rules. For example, if we judge A to be a type and for a free variable $x : A$, we judge $B(x)$ to be a type, then we may judge $(x : A) \to B(x)$ and $\Sigma[\, x \in A \,]\, B(x)$ to be types. The former is the type of functions, and the latter the type of pairs. We define functions $f : (x : A) \to B(x)$ by lambda-abstraction: if $b(x) : B(x)$ for a free variable of type $x : A$, then we judge $\lambda\, x \to b(x) : (x : A) \to B(x)$. We use functions by applying them: if $a : A$ and $f : (x : A) \to B(x)$, then $f(a) : B(a)$, and we take the $(\lambda\, x \to b(x))(a) := b(a)$ and $f := (\lambda\, x \to f(x))$. Similarly, if $a : A$ and $b : B(a)$, we judge $(a, b) : \Sigma[\, x \in A \,]\, B(x)$, and if $p : \Sigma[\, x \in A \,]\, B(x)$ then $\mathsf{fst}\, p : A$ and $\mathsf{snd}\, p : B$; to these we add the defininitional equalities $\mathsf{fst}\, (a, b) := a$, $\mathsf{snd}\, (a, b) := b$, and $p := (\mathsf{fst}\, p, \mathsf{snd}\, p)$.

HoTT extends MLTT with a notion of *path* and an interpretation of types as *homotopy types*. In [5], Awodey and Warren interpret types as *Kan complexes* and elements as simplicial maps (or, more generally, objects and morphisms of a Quillen model category). This is the evident sense in which HoTT gains a homotopy interpretation; but it's worth noting that this model also shows us how HoTT gives us tools for working directly with mathematical structures up to their appropriate notion of equivalence. Kan complexes include not only the singular complexes of topological spaces (their "homotopy types", when considered up to equivalence in the Kan–Quillen model structure), but also the nerves of groupoids of mathematical structures. In the nerve of a groupoid, a path is an isomorphism. By Grothendieck's homotopy hypothesis, Kan complexes are models of ∞-*groupoids*—collections of higher mathematical structures, their equivalences, the equivalences between their equivalences, and so on. Moreover, Lumsdaine [21] and van den Berg and Garner [7] showed that Martin–Löf identity types endow types in MLTT with the structure of (appropriately weak) ∞-groupoids.

https://doi.org/10.1017/jsl.2024.78 Published online by Cambridge University Press

Voevodsky's *univalence principle*, which states that paths in type universes are equivalently equivalences between types, allows us to internalize this understanding of homotopy types as higher groupoids of mathematical structures to some extent. The first set-theoretic model for *univalent* type universes was put forward by Voevodsky and completed by Kapulkin and Lumsdaine [19], interpreting type universes as certain object classifiers in the Kan–Quillen model structure on simplicial sets. We will return to this point later in this introduction; for more on the structural interpretation of HoTT, see [4].

In [28], Shulman shows that HoTT can be interpreted in any ∞-topos. When interpreted in appropriate ∞-toposes (such as stacks on the site of continuous manifolds, as in *ibid.*), ∞-groups are interpreted as stacks of ∞-groups and their deloopings are interpreted as moduli stacks for principal bundles. In this way, we will be able to deduce a Schreier theorem for all stacks, and not just for homotopy types of spaces.

Agda is a dependently typed programming language based on MLTT in which both programs and proofs concerning the behavior of those programs are given by elements of types. Cubical Agda extends Agda with an abstract axiomatization I of the unit interval $[0, 1] \subseteq \mathbb{R}$ and a number of basic operations concerning this interval suitable for doing synthetic homotopy theory in the resulting type theory. See [31] for a more comprehensive introduction to the type theory (originally put forward in [10]), [3] for use in the representation independence of data structures, and [24, 25] for synthetic homotopy theory done in Cubical Agda.

In particular, I has "endpoints" $i0,\ i1 : \mathsf{I}$, a "minimum" function $\_\wedge\_ : \mathsf{I} \to \mathsf{I} \to \mathsf{I}$, a "maximum" function $\_\vee\_ : \mathsf{I} \to \mathsf{I} \to \mathsf{I}$, and a "reversal" $\sim\_ : \mathsf{I} \to \mathsf{I}$ (meant to resemble the function $x \mapsto 1 - x$ on the unit interval) which together equip I with the structure of a de Morgan algebra.

A *path* in Cubical Agda is a function out of the interval I whose value on the endpoints $\mathsf{i0}$ and $\mathsf{i1}$ is known by definition. If $a, b : A$ are elements, then elements of the path type $a \equiv b$ may be defined by λ-abstraction just like functions:

$$\lambda\ i \to p(i) : a \equiv b$$

except that $p(\mathsf{i0}) := a$ and $p(\mathsf{i1}) := b$ by definition. If $A : \mathsf{I} \to \mathsf{Type}$ is a type family varying over the interval, then we also have a type $\mathsf{PathP}\ A\ a\ b$ of paths lying over this path of types, where $a : A\ \mathsf{i0}$ and $b : A\ \mathsf{i1}$.

Paths act as an equality predicate in Cubical Agda. For example, we always have a *reflexivity* term

$$\mathsf{refl} := \lambda\ i \to a : a \equiv a$$

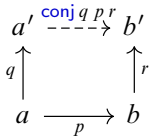
for any $a : A$, as well as symmetry

$$\mathsf{sym} := \lambda\ p\ i \to p(\sim i) : a \equiv b \to b \equiv a.$$

Transitivity may be proven using another primitive notion: *homogenous composition* or hcomp. We will not go into any detail on how hcomp is used here to avoid a digression on partial elements; see [31] for more details. All we will need to know is that hcomp gives us a way to "cap off" open boxes, as in the following definition of

https://doi.org/10.1017/jsl.2024.78 Published online by Cambridge University Press

conjugation of a path $p : a \equiv b$ by paths $q : a \equiv a'$ and $r : b \equiv b'$:



As a special case of conjugation, we get we get transitivity or *path composition* as $p \cdot r := \mathsf{conj}\ \mathsf{refl}\ p\ r$. For those familiar with combinatorial homotopy theory, one can think of hcomp as a Kan-filling operation.

We can prove equations between paths by constructing paths in types of paths; since a path is a function $\mathsf{I} \to A$, a path between paths is a function $\mathsf{I} \to (\mathsf{I} \to A)$ of two interval variables. It is this feature that gives cubical type theory the name; general terms may depend on many interval variables i, j, k which can be thought of as varying over a "cube I^n". We can in particular give paths $p \cdot \mathsf{sym}\ p \equiv \mathsf{refl}$ and $(p \cdot q) \cdot r \equiv p \cdot (q \cdot r)$ and the others needed to show that path composition is unital, invertible, and associative.

With this notion of path, we can define equivalences between types as functions having contractible fibers, following the definition given in the HoTT Book [30].

DEFINITION 2.1. Let $f : A \to B$ be a map and $b : B$. The *fiber* of f over b is the type of elements $a : A$ equipped with a path from $f(a)$ to b:

$$\mathsf{fib}_f(b) := \Sigma[\ a \in A\ ]\ f\ a \equiv b.$$

DEFINITION 2.2. A type A is contractible if when it has a center of contraction $c : A$ and a path from every $x : A$ to c:

$$\mathsf{isContr}(A) := \Sigma[\ c \in A\ ]\ \left((x : A) \to (x \equiv c)\right).$$

A function $f : A \to B$ is an *equivalence* if its fibers are contractible:

$$\mathsf{isEquiv}(f) := (b : B) \to \mathsf{isContr}(\mathsf{fib}_f(b)).$$

We denote the type of equivalences by $A \simeq B$:

$$(A \simeq B) := \Sigma[\ f \in A \to B\ ]\ \mathsf{isEquiv}(f).$$

Since paths are our only form of equality or identification in Cubical Agda, we may think of contractibility as *unique existence*. The principle of unique choice may be understood as the first projection from the type of pairs $\mathsf{isContr}(A)$ to A.

Since paths are functions, it is easy to compute path spaces in pair types $\Sigma[\ a \in A\ ]\ B\ a$:

$$((a, b) \equiv (a', b')) \simeq \Sigma[\ p \in a \equiv a'\ ]\ \mathsf{PathP}\ (\lambda\ i \to B\ (p\ i))\ b\ b'.$$

Similarly, it is easy to compute paths in function types $\Sigma[x : A]B(x)$, giving a form of function extensionality:

$$(f \equiv g) \simeq ((x : A) \to (f(x) \equiv g(x))).$$

Type universes Type should be types as well, and this means that they should implement hcomp for paths (they should be *fibrant*). But what should paths of types be, and how should we compose them? Cubical Agda is *univalent*: paths of types

https://doi.org/10.1017/jsl.2024.78 Published online by Cambridge University Press

Type	Logic	Sets	Homotopy
A	proposition	set	space
$a : A$	proof	element	point
$A + B$	$A \vee B$	disjoint union	coproduct space
$A \times B$	$A \wedge B$	cartesian product	product space

TABLE 1. Correspondence between type theory, logic, sets, and homotopy theory.

correspond to equivalences between them. This is accomplished through another primitive notion: *glue types* Glue. Glue types work like hcomp in that they can "cap off" open boxes of types, but where the base of such a box is a path and the walls are equivalences:

$$\begin{array}{ccc} B_0 & \overset{\mathsf{glue}\, A\, e_0\, e_1}{\dashrightarrow} & B_1 \\ {\scriptstyle e_0}\uparrow & & \uparrow{\scriptstyle e_1} \\ A\, i_0 & \xrightarrow[A]{} & A\, i_1 \end{array}$$

By gluing the reflexive path onto an equivalence, we get Voevodsky's *univalence* principle: paths between types are equivalences:

$$\begin{array}{ccc} B_0 & \overset{\mathsf{ua}\, e}{\dashrightarrow} & A \\ {\scriptstyle e}\uparrow & & \uparrow{\scriptstyle \mathsf{id}} \\ A & \xrightarrow[\mathsf{refl}]{} & A \end{array}$$

With glue types and univalence, we can construct paths $p : A \equiv B$ between types by constructing equivalences $e : A \simeq B$ between them; composition of paths corresponds to composition of their corresponding equivalences. In particular, the type $A \equiv A$ of *loops* starting and ending at a type A is equivalent to the type $\mathsf{Aut}(A)$ of automorphisms of A.

Being contractible is a property of types, and being an equivalence is a property of functions. In type theory, propositions are represented as types, a practice known as *propositions as types*. The judgement $a : A$ can mean that a is a mathematical object of type A and also that a is a witness to the truth of the proposition A. We summarize the different viewpoints of the HoTT operations in Table 1.

As with other dependently typed languages, in Cubical Agda propositions are themselves expressed as types. However, following practice in HoTT [30], we do not consider all types to be propositions—some types are sets, like $\mathbb{N}$, and some are types of mathematical structures, like the type of groups, for example. We identify the propositions as those types that have at most one element.

DEFINITION 2.3. A type A is a proposition when for any two elements $a, b : A$, the type of paths $(a \equiv b)$ between them is contractible:

$$\mathsf{isProp}(A) := (a\, b : A) \to \mathsf{isContr}(a \equiv b).$$

https://doi.org/10.1017/jsl.2024.78 Published online by Cambridge University Press

If $P : A \to \mathsf{Type}$ is a family of propositions (that is, for all $a : A$, $\mathsf{isProp}(P\,a)$), then the pair type $\Sigma[\ a \in A\ ]\ P\,a$ acts as the subtype of A consisting of those elements which satisfy P. By the computation of paths in pair types and the fact that paths in propositions are trivial, paths in subtypes may be computed in the underlying type:

$$((a, p) \equiv (a', p')) \simeq (a \equiv a).$$

Cubical Agda has *higher inductive types* (HITs), or inductive types with path constructors as well as term constructors. We may define the *propositional truncation* or *squash type* $||A||$ given by the following recursive HIT:

```
data ∥_∥ (A : Type) : Type where
  trunc : A → ∥ A ∥
  squash : (a b : ∥ A ∥) → a ≡ b
```

Giving an element of $||A||$ proves the proposition that there is *some* element of A, though there is not in general a function $||A|| \to A$ which lets us extract a specific witness. Propositional truncation therefore acts as an existential quantifier in cubical Agda: $||A||$ asserts that there *exists* an element of A, though it does not provide a specific witness to that existence.

Following the HoTT Book [30], we may define a type A to be a *set* if the path types in A are propositions:

$$\mathsf{isSet}(A) := (a\ b : A) \to \mathsf{isProp}(a \equiv b).$$

Thinking of paths as equality predicates, this means that a set is a type in which equality is a proposition, as it usually considered to be.

With the type constructors so far described, we can now give an example of a type of mathematical structures. If G is a group, then we can define the type of left G-torsors (left G actions which are free, transitive, and inhabited) as follows:

$$\mathsf{Tors}_G := \begin{cases} \Sigma[\ T \in \mathsf{Type}\ ]\ \mathsf{isSet}(T) \\ \times\ \Sigma[\ \alpha \in G \to (T \to T)\ ] \\ \times\ (t : T) \to (\alpha(1, t) \equiv t) \\ \times\ (g, h : G) \to (t : T) \to (\alpha(gh, t) \equiv \alpha(g, \alpha(h, t))) \\ \times\ (t : T) \to \mathsf{isEquiv}(\lambda\ g \to \alpha(g, t)) \\ \times\ |T| \end{cases}$$

Since paths in pairs may be computed componentwise, and since paths in propositions may be ignored (and types of paths in sets are propositions), we can compute that the type of paths $(T, \ldots) \equiv (T', \ldots)$ is equivalent to the type of G-equivariant isomorphisms between T and T'. Generally speaking, the *structure identity principle* (SIP) (see, e.g., [1]) guarantees that paths in types of mathematical structures correspond to structure-preserving equivalences In particular, the type of *loops* $(G, \ldots) \equiv (G, \cdot)$ on G considered as a torsor over itself by left multiplication is equivalent to the type of G-equivariant automorphisms of G, which is equivalent to the group G itself. We will use the SIP to interpret group theory as the theory of symmetries of mathematical structures in the following section.

https://doi.org/10.1017/jsl.2024.78 Published online by Cambridge University Press

§3. Higher groups. In this section, we will provide the necessary background for the higher Schreier theorem. We begin by revisiting the theory of higher groups as laid out in [9].

DEFINITION 3.1. A pointed type is type A equipped with a point $a : A$. We will usually refer to the pointed type by the name of the type A and its basepoint by pt_A.

$$\mathsf{Ptd} := \Sigma[\ A \in \mathsf{Type}\ ]\ A$$

$$\mathsf{pt} : ((A, a) : \mathsf{Ptd}) \to A$$
$$\mathsf{pt}(A, a) := a.$$

A pointed function $(f, \mathsf{pt}_f) : A \to_* B$ is a function $f : A \to B$ that preserves the basepoints $\mathsf{pt}_f : f(\mathsf{pt}_A) \equiv \mathsf{pt}_B$. We define the pointed composite by

$$(g, \mathsf{pt}_g) \circ_* (f, \mathsf{pt}_f) := (g \circ f, (\mathsf{cong}\, g\, \mathsf{pt}_f) \cdot \mathsf{pt}_g)$$

DEFINITION 3.2. For $A : \mathsf{Ptd}$, its *loop space* $\Omega A : \mathsf{Ptd}$ is the type of paths from pt_A to itself, pointed at reflexivity:

$$\Omega A := ((\mathsf{pt}_A \equiv \mathsf{pt}_A), \mathsf{refl}).$$

Loops Ω acts as a functor $\mathsf{Ptd} \to \mathsf{Ptd}$:

$$\Omega f := \lambda\ p \to \mathsf{sym}\, \mathsf{pt}_f\ \cdot\ \mathsf{cong}\, f\ p\ \cdot\ \mathsf{pt}_f$$

$$\mathsf{pt}_{\Omega f} := \mathsf{refl}$$

For any pointed type A, the type $\Omega A := (\mathsf{pt}_A \equiv \mathsf{pt}_A)$ has a group-like structure given by path composition. If pt_A is essentially the only element in A (up to paths), then this group-like structure ΩA even determines A in the following way. As we saw in the previous section, the structure identity principle lets us identify the loop space of a type of mathematical structures with with the automorphism group of a canonical example of that structure.

A type is said to be 0-connected if there exists a path between any two elements:

$$\mathsf{is0Connected}(A) := (a\ b : A) \to ||a \equiv b||.$$

Unlike a contractible type where there is a contractible type of paths between any two elements, the path type $a \equiv b$ between elements of a 0-connected type may be highly non-trivial. We will exploit these non-trivial path types to give an account of groups as paths in a 0-connected type.

DEFINITION 3.3. A type A is 0-connected if there is merely a path between any two points of A:

$$\mathsf{is0Connected}(A) := (a\ b : A) \to ||a \equiv b||.$$

We denote by $\mathsf{Type}_{>0}$ the type of 0-connected types.

Thinking of the type A as a type of mathematical structures, A is 0-connected when the theory of that structure is *categorical*—there is a single model up to (not necessarily unique) isomorphism. As an example, the type of algebraically closures of a field is 0-connected; picking a canonical algebraic closure, the loop space of the

https://doi.org/10.1017/jsl.2024.78 Published online by Cambridge University Press

type of algebraic closures is then, by the structure identity principle, the absolute Galois group of the field.

LEMMA 3.4. *A pointed function* $i : A \to_* B$ *between pointed* 0*-connected types is an equivalence if and only if* $\Omega i : \Omega A \to_* \Omega B$ *is.*

PROOF. If f is an equivalence, it is straightforward to show that Ωf is an equivalence—this does not require 0-connectedness of A and B.

For the converse, we aim to show that the fibers $\mathsf{fib}_f(b)$ of f are contractible for all $b : B$ on the assumption that Ωf is an equivalence. Since being contractible is a proposition and B was assumed 0-connected, it will suffice to show that the fiber $\mathsf{fib}_f(\mathsf{pt}_B)$ over the basepoint is contractible.

$$\mathsf{fib}_f(\mathsf{pt}_B) := \Sigma[\ a \in A\ ]\ (f(a) \equiv \mathsf{pt}_B).$$

To show that $\mathsf{fib}_f(\mathsf{pt}_B)$ is contractible, it will therefore suffice to show that for all $a : A$, $(f(a) \equiv \mathsf{pt}_B) \simeq (a \equiv \mathsf{pt}_A)$, since singleton types $\Sigma[\ a \in A\ ]\ (a \equiv \mathsf{pt}_A)$ are contractible. Note that for any $a : A$ we have a map

$$\lambda\ p.\, (\mathsf{cong}\ f, p) \cdot \mathsf{pt}_f : (a \equiv \mathsf{pt}_A) \to (f(a) \equiv \mathsf{pt}_B).$$

Showing that this map is an equivalence is a proposition, and since A is 0-connected we may prove this proposition in the case that a is pt_A. But in this case, we have a commuting triangle as follows by the definition of Ωf:

$$\begin{array}{ccc} (\mathsf{pt}_A \equiv \mathsf{pt}_A) & \xrightarrow{\lambda\ p.\ (\mathsf{cong}\ f\ p) \cdot \mathsf{pt}_f} & (f(\mathsf{pt}_A) \equiv \mathsf{pt}_B) \\ & \searrow^{\Omega f} & \downarrow{\scriptstyle \lambda\ p.\ (\mathsf{sym}\ \mathsf{pt}_f) \cdot p} \\ & & (\mathsf{pt}_B \equiv \mathsf{pt}_B) \end{array}$$

Since Ωf was assumed to be an equivalence, and since composition with a path is an equivalence, we conclude that $\lambda\ p.\, (\mathsf{cong}\ f\ p) \cdot \mathsf{pt}_f$ is an equivalence and therefore that f is as well. ⊣

For this reason, pointed connected types are determined by their loop spaces. Following [9], we adopt the perspective that these loop spaces are *higher groups*, and that the associated pointed connected types are their *deloopings*.

DEFINITION 3.5. A *higher group* is a pointed, 0-connected type $\mathsf{B}G$. We refer to $G := \Omega\mathsf{B}G$ as the group itself, while $\mathsf{B}G$ is its "delooping". A (1-)group is a higher group $\mathsf{B}G$ for which $\mathsf{B}G$ is 1-truncated, or equivalently for which $G := \Omega\mathsf{B}G$ is a set.

A *homomorphism* $\varphi : G \to H$ between higher groups is a pointed function $\mathsf{B}\varphi : \mathsf{B}G \to_* \mathsf{B}H$ between their deloopings.

Higher groups are *concrete groups*: they are directly represented as the symmetries of a given object (namely, the base point of their delooping). As we saw above, any This contrasts with *abstract groups* presented axiomatically as sets equipped with the usual operations.

These two approaches to group theory agree in the case of 1-groups: for any concrete 1-group with delooping $\mathsf{B}G$, we may define an abstract group structure on $G := \Omega\mathsf{B}G$ using composition of paths. By the functoriality of Ω, a homomorphism

https://doi.org/10.1017/jsl.2024.78 Published online by Cambridge University Press

$\mathsf{B}\varphi : \mathsf{B}G \to \mathsf{B}H$ induces a homomorphism of abstract groups $\varphi := \Omega\mathsf{B}\varphi : G \to_* H$ since *any* function in Cubical Agda commutes with composition of paths. This functor gives an between the category of concrete 1-groups and the category abstract 1-groups. A formalization of this equivalence is beyond the scope of this paper. We will take for granted that any abstract group G can be delooped by a pointed 0-connected type $\mathsf{B}G$ (necessarily 1-truncated, since G must be a set), and that homomorphisms between abstract groups are equivalently given by pointed maps between their deloopings. For a full proof of this result, see [2]; see also [20] for a construction of deloopings using higher inductive types.

From now on, we will work only with concrete (higher) groups defined as in Definition 3.5, and will drop the modifier "concrete". The easiest way to construct examples of higher groups is to take the groups of symmetries of a given element of a type.

Definition 3.6. Let A be a type and $a : A$ an element. Define

$$\mathsf{BAut}_A(a) := \Sigma[\ x \in A\ ]\ \|x \equiv a\|$$

to be the type of elements of x which are path connected to a, and define $\mathsf{pt}_{\mathsf{BAut}_A(a)} := (a, |\mathsf{refl}|)$.

It follows quickly by the computation of path in subtypes that $\Omega\mathsf{BAut}_A(a) \simeq (a \equiv a)$ and that $\mathsf{BAut}_A(a)$ is 0-connected. We will also make the following special definitions in the case that A is a type universe.

Definition 3.7. Let X be a type. We define

$$\mathsf{BAut}(X) := \mathsf{BAut}_{\mathsf{Type}}(X) \simeq \Sigma[\ Y \in \mathsf{Type}\ ]\ \|Y \simeq X\|.$$

and if $\mathsf{pt}_X : X$ is a base point, we define

$$\mathsf{BAut}_*(X) := \mathsf{BAut}_{\mathsf{Ptd}}((X, \mathsf{pt}_X))$$

By univalence, $\mathsf{BAut}(X)$ is a delooping of the type of self-equivalences $\mathsf{Aut}(X) := (X \simeq X)$. Note that since homomorphisms $G \to H$ between higher groups are pointed maps $\mathsf{B}G \to_* \mathsf{B}H$ between their deloopings, $\mathsf{BAut}_*(\mathsf{B}G)$ deloops the higher group of *higher group automorphisms* $\mathsf{Aut}_*(\mathsf{B}G)$ of G. For more examples of concrete deloopings, see [22].

§4. The higher Schreier theorem. In this section, we will prove the higher Schreier theorem. First, we must define extensions of higher groups. To do this, we will need the general notion of a *fiber sequence*.

Definition 4.1. Let A, B, and C be pointed types. A *fiber sequence* $A \to_* B \to_* C$ consists of a pointed function $f : B \to_* C$ together with a pointed equivalence $e : A \simeq_* \mathsf{fib}_f$. The map $A \to_* B$ suggested by the notation $A \to_* B \to_* C$ is defined to be the composite $A \simeq_* \mathsf{fib}_f \to_* B$ of e with the first projection $\mathsf{fib}_f \to_* B$.

Definition 4.2. Let G and K be higher groups. An *extension* of G by K consists of a fiber sequence $\mathsf{B}K \to_* \mathsf{B}E \to_* \mathsf{B}G$ with $\mathsf{B}E$ a 0-connected type. We denote the type of extensions by $\mathsf{EXT}(G, K)$.

https://doi.org/10.1017/jsl.2024.78 Published online by Cambridge University Press

LEMMA 4.3. *In the definition of extension, we do not have to assume that* $\mathsf{B}E$ *is* 0*-connected*:

$$\mathsf{EXT}(G,K) \simeq \Sigma[\ \mathsf{B}E \in \mathsf{Ptd}\ ]\ \Sigma[\ \mathsf{B}p \in \mathsf{B}E \to_* \mathsf{B}G\ ]\ (\mathsf{B}K \simeq_* \mathsf{fib}_{\mathsf{B}p})$$

PROOF. Suppose we have $\mathsf{B}p : \mathsf{B}E \to_* \mathsf{B}G$ with $\mathsf{B}K \simeq_* \mathsf{fib}_{\mathsf{B}p}$ as on the right. Since $\mathsf{B}G$ is 0-connected, all fibers of $\mathsf{B}p$ are merely equivalent to the fiber over the base point, which by assumption is equivalent to $\mathsf{B}K$. Therefore, $\mathsf{B}E$ is equivalent to the sum of 0-connected types, indexed by a 0-connected type $\mathsf{B}G$, and so it itself 0-connected. ⊣

Our proof of the higher Schreier theorem relies on a general lemma. Recall that for any type A there is an equivalence $(A \to \mathsf{Type}) \simeq \Sigma[\ B \in \mathsf{Type}\ ]\ (B \to A)$ of type families on A and functions into A given by taking Σ-types in one direction and taking fibers in the other [30]. We can extend this equivalence to one for pointed types.

DEFINITION 4.4. Let A be a pointed type. A *pointed type family* on A is a type family $B : A \to_* \mathsf{Type}$ together with a point $\mathsf{pt}_B : B\ \mathsf{pt}_A$ over the base point of A.

$$\mathsf{Ptd}_A := \Sigma[\ B \in A \to \mathsf{Type}\ ]\ B\ \mathsf{pt}_A.$$

LEMMA 4.5. *There is an equivalence*

$$\mathsf{Ptd}_A \simeq \Sigma[\ B \in \mathsf{Ptd}\ ]\ (B \to_* A)$$

between pointed families on A *and pointed functions into* A.

PROOF.

$$\begin{aligned}
\mathsf{Ptd}_A &:= \Sigma[\ B \in A \to \mathsf{Type}\ ]\ B\ \mathsf{pt}_A \\
&\simeq \Sigma[\ B \in \mathsf{Type}\ ]\ \Sigma[\ f \in B \to A\ ]\ \mathsf{fib}_f(\mathsf{pt}_A) \\
&\simeq \Sigma[\ B \in \mathsf{Type}\ ]\ \Sigma[\ f \in B \to A\ ]\ \Sigma[\ \mathsf{pt}_B \in B\ ]\ (f(b) \equiv \mathsf{pt}_A) \\
&\simeq \Sigma[\ B \in \mathsf{Ptd}\ ]\ (B \to_* A)
\end{aligned}$$

⊣

With this lemma in hand, we may derive the higher Schreier theorem.

THEOREM 4.6. *Let* G *and* K *be higher groups. Then*

$$\mathsf{EXT}(G,K) \simeq (\mathsf{B}G \to_* \mathsf{BAut}(\mathsf{B}K)).$$

PROOF.

$$\begin{aligned}
\mathsf{EXT}(G,K) &\simeq \Sigma[\ \mathsf{B}E \in \mathsf{Ptd}\ ]\ \Sigma[\ \mathsf{B}p \in \mathsf{B}E \to_* \mathsf{B}G\ ]\ (\mathsf{B}K \equiv_{\mathsf{Ptd}} \mathsf{fib}_{\mathsf{B}p}) \\
&\simeq \begin{Sigma}[\ c \in \mathsf{B}G \to \mathsf{Type}\ ]\ \Sigma[\ \mathsf{pt}_c \in c(\mathsf{pt}_{\mathsf{B}G})\ ] \\ \Sigma[\ e \in c(\mathsf{pt}_{\mathsf{B}G}) \simeq \mathsf{B}K\ ]\ (e(\mathsf{pt}_c)) \equiv \mathsf{pt}_{\mathsf{B}K}\end{Sigma} \\
&\simeq \begin{Sigma}[\ c \in \mathsf{B}G \to \mathsf{Type}\ ]\ \Sigma[\ \mathsf{pt}_c \in c(\mathsf{pt}_{\mathsf{B}G})\ ] \\ \Sigma[\ e \in c(\mathsf{pt}_{\mathsf{B}G}) \simeq \mathsf{B}K\ ]\ (\mathsf{pt}_c \equiv e^{-1}\mathsf{pt}_{\mathsf{B}K})\end{Sigma} \\
&\simeq \Sigma[\ c \in \mathsf{B}G \to \mathsf{Type}\ ]\ (c(\mathsf{pt}_{\mathsf{B}G}) \simeq \mathsf{B}K) \\
&\simeq \mathsf{B}G \to_* \mathsf{BAut}(\mathsf{B}K)
\end{aligned}$$

https://doi.org/10.1017/jsl.2024.78 Published online by Cambridge University Press

The last equivalence follows because $\mathsf{B}G$ is 0-connected; since $c(\mathsf{pt}_{\mathsf{B}G}) \simeq \mathsf{B}K$, it follows that $||c(t) \simeq \mathsf{B}K||$ for all $t : \mathsf{B}G$ since $||t \equiv \mathsf{pt}_{\mathsf{B}G}||$ by hypothesis. ⊣

§5. Classifying ordinary extensions. In this section, we will prove that extensions of higher groups defined as in Definition 4.2 generalize extensions of ordinary (1-)groups. We will organize the results in this section in terms of squares of pointed types.

DEFINITION 5.1. A square S of pointed types consists of:

$$S := \begin{array}{ccc} A & \xrightarrow{i} & B \\ {\scriptstyle q}\downarrow & & \downarrow{\scriptstyle p} \\ C & \xrightarrow[j]{} & D \end{array}$$

1. Pointed types A, B, C, D;
2. Pointed maps $i : A \to_* B$, $j : C \to_* B$, $q : A \to_* C$ and $p : B \to_* D$;
3. A path $S : p \circ_* i \equiv j \circ_* q$ in the type of pointed maps $A \to_* D$.

DEFINITION 5.2. Given a square of pointed types S, define its loops ΩS to be the square

$$\Omega S := \begin{array}{ccc} \Omega A & \xrightarrow{\Omega i} & \Omega B \\ {\scriptstyle \Omega q}\downarrow & & \downarrow{\scriptstyle \Omega p} \\ \Omega C & \xrightarrow[\Omega j]{} & \Omega D \end{array}$$

We need a general theorem relating squares of pointed types and their loopings.

THEOREM 5.3. *Let S be a square of pointed types*:

$$S := \begin{array}{ccc} A & \xrightarrow{i} & B \\ {\scriptstyle q}\downarrow & & \downarrow{\scriptstyle p} \\ C & \xrightarrow[j]{} & D \end{array}$$

1. *If S is a pullback, then so is ΩS.*
2. *If C, q and p are 0-connected, then if ΩS is a pullback, so is S.*

PROOF. The first implication may be proven directly. Note that it also implies that $\Omega\mathsf{fib}_p \simeq \mathsf{fib}_{\Omega p}$ for any pointed map $p : B \to_* D$ by taking C to be the unit type.

For the second implication, recall that a square S is a pullback if and only if the induced map $S_* : \mathsf{fib}_q(c) \to \mathsf{fib}_p(j(c))$ is an equivalence for all $c : C$. If C is 0-connected, it suffices to consider $S_* : \mathsf{fib}_q(\mathsf{pt}_C) \to \mathsf{fib}_p(j(\mathsf{pt}_C))$, which is a pointed map between pointed types because both q and p are pointed maps and by the commutativity of the square. If q and p are 0-connected, then $\mathsf{fib}_q(\mathsf{pt}_C)$ and $\mathsf{fib}_p(j(\mathsf{pt}_C))$ are both 0-connected, and so by Lemma 3.4 it follows that S_* is an equivalence whenever $\Omega S_* : \Omega\mathsf{fib}_q(\mathsf{pt}_C) \to_* \Omega\mathsf{fib}_p(j(\mathsf{pt}_C))$ is. But as a corollary of the first implication, this map is equivalent to $(\Omega S)_* : \mathsf{fib}_{\Omega q}(\mathsf{refl}_{\mathsf{pt}_C}) \to \mathsf{fib}_{\Omega p}(\mathsf{refl}_{j(\mathsf{pt}_C)})$. If ΩS is a pullback, then $(\Omega S)_*$ is an equivalence, proving the desired implication. ⊣

https://doi.org/10.1017/jsl.2024.78 Published online by Cambridge University Press

We are now ready to prove the equivalence between extensions as classified in the higher Schreier theorem and the classical notion of extensions between groups. We begin by recalling the classical definition of extension of 1-groups as a short exact sequence.

DEFINITION 5.4. Let G and K be groups. An *extension* of G by K is a short exact sequence

$$0 \to K \xrightarrow{i} E \xrightarrow{p} G \to 0$$

Explicitly, this consists of a group E together with homomorphisms $i : K \to E$ and $p : E \to G$ such that:

1. $p \circ i$ is constant at $1 : G$,
2. i is injective,
3. p is surjective,
4. The induced map $K \to \ker(p)$ is an equivalence.

By the definition of groups and group homomorphisms Definition 3.5, the above description of a short exact sequence is by definition equivalent to the following. A short exact sequence is a square of pointed types:

$$S := \begin{array}{ccc} \mathsf{B}K & \xrightarrow{\mathsf{B}i} & \mathsf{B}E \\ {\scriptstyle !}\downarrow & & \downarrow{\scriptstyle \mathsf{B}p} \\ * & \xrightarrow[\mathsf{pt}_B]{} & \mathsf{B}G \end{array}$$

with $\mathsf{B}K$, $\mathsf{B}E$, and $\mathsf{B}G$ 0-connected such that

1. $i := \Omega\mathsf{B}i$ is injective (– 1-truncated),
2. $p := \Omega\mathsf{B}p$ is surjective (– 1-connected),
3. and ΩS is a pullback; equivalently, the induced map $K \to \ker p$ is an equivalence.

To compare the two notions of extension, we begin by putting the definition of extension from Definition 4.2 in a similar form.

LEMMA 5.5. *Let G and K be higher groups. Extensions of G by K are equivalently pointed pullback squares*

$$S := \begin{array}{ccc} \mathsf{B}K & \xrightarrow{\mathsf{B}i} & \mathsf{B}E \\ {\scriptstyle !}\downarrow & & \downarrow{\scriptstyle \mathsf{B}p} \\ * & \xrightarrow[\mathsf{pt}_B]{} & \mathsf{B}G \end{array}$$

PROOF. It is generally true that fiber sequences are equivalent to pullback squares of the above sort, since the fiber is the pullback of a point. ⊣

We may then put these lemmas together with the help of Theorem 5.3.

THEOREM 5.6. *Let G and K be groups. Then the type of extensions of G by K is equivalent to the type of short exact sequences $K \to E \to G$.*

https://doi.org/10.1017/jsl.2024.78 Published online by Cambridge University Press

PROOF. By appealing to Lemma 5.5, it suffices to show that for any square of pointed 0-connected types

$$S := \quad \begin{array}{ccc} \mathsf{B}K & \xrightarrow{\mathsf{B}i} & \mathsf{B}E \\ {\scriptstyle !}\downarrow & & \downarrow{\scriptstyle \mathsf{B}p} \\ * & \xrightarrow[\mathsf{pt}_B]{} & \mathsf{B}G \end{array}$$

S is a pullback if and only if ΩS is a pullback, $i := \Omega \mathsf{B} i$ is injective, and $p := \Omega \mathsf{B} p$ is surjective. If S is a pullback, then ΩS is a pullback by Theorem 5.3; i is injective because $\mathsf{B}i$ has the 0-truncated fiber G; p is surjective because $\mathsf{B}p$ has 0-connected fiber $\mathsf{B}K$. Conversely, if p is surjective then $\mathsf{B}p$ is 0-connected, and since $\mathsf{B}K$ is also 0-connected, Theorem 5.3 again shows that if ΩS is a pullback, so is S. ⊣

§6. Split extensions and semi-direct products. In this section, we will recover the classical classification of split extensions by homomorphic group actions. Recall that $\mathsf{BAut}_*(\mathsf{B}K)$ deloops the type $\mathsf{Aut}_*(\mathsf{B}K)$ of pointed automorphisms of $\mathsf{B}K$, or equivalently higher group automorphisms of K. Therefore, a homomorphic action of G on K—that is, an action of G on K via homomorphisms—is equivalently a pointed map $\mathsf{B}G \to_* \mathsf{BAut}_*(\mathsf{B}K)$. We will show that such pointed maps classify split extensions.

We begin by defining the *holomorph* of a higher group K, a classical notion (for 1-groups) which is in a precise sense the universal semi-direct product with K.

DEFINITION 6.1. Consider the function $\pi : \mathsf{BAut}_*(\mathsf{B}K) \to_* \mathsf{BAut}(\mathsf{B}K)$ which sends a pointed type (A, pt_A) merely equivalent to $(\mathsf{B}K, \mathsf{pt}_{\mathsf{B}K})$ to its underlying type A. By the higher Schreier theorem, this corresponds to an extension:

$$\mathsf{B}K \to_* \mathsf{BHol}(K) \to_* \mathsf{BAut}_*(\mathsf{B}K)$$

of $\mathsf{Aut}_*(\mathsf{B}K)$ by K where

$$\mathsf{BHol}(K) \simeq \Sigma[\, A \in \mathsf{Ptd} \,]\, A \times \|A \simeq \mathsf{B}K\|$$

is equivalently the type of doubly pointed types merely equivalent to $\mathsf{B}K$. This extension is called the *holomorph* of K.

We will show that an extension is split if and only if its classifying map factors through the map $\pi : \mathsf{BAut}_*(\mathsf{B}K) \to_* \mathsf{BAut}(\mathsf{B}K)$ classifying the holomorph of K.

First, we need an abstract lemma related pointed sections to pointed dependent functions.

DEFINITION 6.2. Let $B : \mathsf{Ptd}_A$ be a pointed family of types on A. The type of *pointed dependent functions* is defined as follows.

$$((a : A) \to_* B\, a) := \Sigma[\, f \in (a : A) \to B\, a \,]\, (f(\mathsf{pt}_A) \equiv \mathsf{pt}_B)$$

If $f : B \to_* A$ is a pointed function, then a pointed section of f is a pointed function $s : A \to_* B$ together with a path $f \circ_* s \equiv (\mathrm{id}_A, \mathsf{refl}_{\mathsf{pt}_A})$ in the type of pointed functions $A \to_* A$.

https://doi.org/10.1017/jsl.2024.78 Published online by Cambridge University Press

LEMMA 6.3. *Let A be a pointed type. The equivalence of Lemma 4.5 between pointed families on A and pointed functions into A extends to an equivalence between pointed dependent functions of a pointed family on A and pointed sections of a pointed function into A.*

Now we define split exact sequences of higher groups and prove our desired classification.

DEFINITION 6.4. Let $\mathsf{B}K \xrightarrow{\mathsf{B}i}_* \mathsf{B}E \xrightarrow{\mathsf{B}p}_* \mathsf{B}G$ be an extension of higher groups. A *splitting* of this extension is a pointed map $\mathsf{B}s : \mathsf{B}G \to_* \mathsf{B}E$ together with a path $\mathsf{B}p \circ_* \mathsf{B}s \equiv (\mathrm{id}_{\mathsf{B}p}, \mathsf{refl}_{\mathsf{pt}_{\mathsf{B}G}})$ in the type of pointed maps $\mathsf{B}G \to_* \mathsf{B}G$.

A *split extension* is an extension with a choice of splitting.

THEOREM 6.5. *Let $\mathsf{B}K \xrightarrow{\mathsf{B}i}_* \mathsf{B}E \xrightarrow{\mathsf{B}p}_* \mathsf{B}G$ be an extension of higher groups classified by a map $c : \mathsf{B}G \to_* \mathsf{BAut}(\mathsf{B}K)$. Then splittings of this extension correspond to pointed factorizations of c through $\pi : \mathsf{BAut}_*(\mathsf{B}K) \to \mathsf{BAut}(\mathsf{B}K)$.*

PROOF. We may consider c to be a pointed family by transporting $\mathsf{pt}_{\mathsf{B}K}$ over $\mathsf{pt}_c : c(\mathsf{pt}_{\mathsf{B}G}) \equiv \mathsf{B}K$. By Lemma 6.3, splittings of the extension are therefore equivalent to pointed dependent functions $(t : \mathsf{B}G) \to_* c(t)$. It remains to show that

$$(t : \mathsf{B}G) \to_* c(t) \simeq \Sigma[\, \tilde{c} \in \mathsf{B}G \to_* \mathsf{BAut}_*(\mathsf{B}K)\,]\ (\pi \circ_* \tilde{c} \equiv c).$$

We compute as follows:

$$\begin{aligned}
&\Sigma[\, \tilde{c} \in \mathsf{B}G \to_* \mathsf{BAut}_*(\mathsf{B}K)\,]\ (\pi \circ_* \tilde{c} \equiv c) \\
&\simeq \begin{cases} \Sigma[\, \tilde{c} \in \mathsf{B}G \to \mathsf{BAut}_*(\mathsf{B}K)\,]\ \Sigma[\, \mathsf{pt}_{\tilde{c}} \in \tilde{c}(\mathsf{pt}_{\mathsf{B}G}) \equiv (\mathsf{B}K, \mathsf{pt}_{\mathsf{B}K})\,] \\ \quad \Sigma[\, p \in \pi \circ \tilde{c} \equiv c\,]\ ((\lambda\, i \to p\, i\, \mathsf{pt}_{\mathsf{B}K}) \cdot \mathsf{pt}_c \equiv \mathsf{cong}\, \pi\, \mathsf{pt}_{\tilde{c}}) \end{cases} \\
&\simeq \begin{cases} \Sigma[\, s \in (t : \mathsf{B}G) \to c(t)\,] \\ \quad \Sigma[\, \mathsf{pt}_{\tilde{c}} \in (c(\mathsf{pt}_{\mathsf{B}G}), s(\mathsf{pt}_{\mathsf{B}G})) \equiv (\mathsf{B}K, \mathsf{pt}_{\mathsf{B}K})\,] \\ \quad (\mathsf{pt}_c \equiv \mathsf{cong}\, \pi\, \mathsf{pt}_{\tilde{c}}) \end{cases} \\
&\simeq \Sigma[\, s \in (t : \mathsf{B}G) \to c(t)\,]\ (\mathsf{transport}\, \mathsf{pt}_c\, (s\, \mathsf{pt}_{\mathsf{B}G}) \equiv \mathsf{pt}_{\mathsf{B}K}) \\
&\simeq \Sigma[\, s \in (t : \mathsf{B}G) \to c(t)\,]\ (s\, \mathsf{pt}_{\mathsf{B}G} \equiv \mathsf{transport}\, (\mathsf{sym}\, \mathsf{pt}_c)\, \mathsf{pt}_{\mathsf{B}K})
\end{aligned}$$

⊣

REFERENCES

[1] B. AHRENS and P. R. NORTH, *Univalent foundations and the equivalence principle*, ***Reflections on the Foundations of Mathematics—Univalent Foundations, Set Theory and General Thoughts*** (S. Centrone, D. Kant, and D. Sarikaya, editors), ***vol.*** 407, Synthese Library, Springer, Cham, 2019, pp. 137–150.

[2] M. BEZEM, U. BUCHHOLTZ, P. CAGNE, B. I. DUNDAS, and D. R. GRAYSON, *Symmetry*. Open draft at https://github.com/UniMath/SymmetryBook.

[3] C. ANGIULI, E. CAVALLO, A. MÖRTBERG, and M. ZEUNER, *Internalizing representation independence with univalence*. ***Proceedings of the ACM on Programming Languages***, vol. 5 (2021), pp. 1–30. https://doi.org/10.1145/3434293.

[4] S. AWODEY, *Structuralism, invariance, and univalence*. ***Philosophia Mathematica***, vol. 22 (2014), no. 1, pp. 1–11.

[5] S. AWODEY and M. A. WARREN, *Homotopy theoretic models of identity types*. ***Mathematical Proceedings of the Cambridge Philosophical Society***, vol. 146 (2009), no. 1, pp. 45–55.

https://doi.org/10.1017/jsl.2024.78 Published online by Cambridge University Press

[6] R. Baer, *Erweiterung von Gruppen und ihren Isomorphismen*. ***Mathematische Zeitschrift***, vol. 38 (1934), no. 1, pp. 375–416.

[7] B. van den Berg and R. Garner, *Types are weak ω-groupoids*. ***Proceedings of the London Mathematical Society***, vol. 102 (2011), no. 2, pp. 370–394.

[8] L. Breen, *Théorie de Schreier supérieure*. ***Annales Scientifiques de l'École Normale Supérieure***, vol. 25 (1992), no. 5, pp. 465–514.

[9] U. Buchholtz, F. van Doorn, and E. Rijke, *Higher groups in homotopy type theory*, ***LICS '18—33rd Annual ACM/IEEE Symposium on Logic in Computer Science***, ACM, New York, 2018, p. 10.

[10] C. Cohen, C. Thierry, H. Simon, and M. Anders, *Cubical Type Theory: a constructive interpretation of the univalence axiom*, ***21st International Conference on Types for Proofs and Programs***. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Tallinn, 2015, p. 262.

[11] P. Dedecker, *Les foncteurs Ext_Π, $H_\Pi{}^2$ et $H_\Pi{}^2$ non abéliens*. ***Comptes Rendus de l'Académie des Sciences***, vol. 258 (1964), pp. 4891–4894.

[12] P. Dedecker and E. M. Luks, *Sur la non-fonctorialité du H^2 en cohomologie non abélienne*. ***Comptes Rendus de l'Académie des Sciences***, vol. 282 (1976), no. 3, Ai, A139–A141.

[13] S. Eilenberg and S. MacLane, *Cohomology theory of Abelian groups and homotopy theory. I*. ***Proceedings of the National Academy of Sciences of the United States of America***, vol. 36 (1950), pp. 443–447.

[14] ———, *Cohomology theory of Abelian groups and homotopy theory. II*. ***Proceedings of the National Academy of Sciences of the United States of America***, vol. 36 (1950), pp. 657–663.

[15] ———, *Cohomology theory of Abelian groups and homotopy theory. III*. ***Proceedings of the National Academy of Sciences of the United States of America***, vol. 37 (1951), pp. 307–310.

[16] J. Giraud, ***Cohomologie non abélienne***, Die Grundlehren der Mathematischen Wissenschaften, 179, Springer-Verlag, Berlin, 1971, pp. ix+467.

[17] A. Grothendieck, ***Rev ê tements é tales et groupe fondamental. Fasc. I: Expos é s 1 à 5***, Troisième éd., corrigée, Séminaire de Géométrie Algébrique, 1960/61, Institut des Hautes Études Scientifiques, Paris, 1963, pp. iv+143. (not consecutively paged) (loose errata).

[18] M. Hofmann and T. Streicher, *The groupoid interpretation of type theory*, ***Twenty-Five Years of Constructive Type Theory (Venice, 1995)***, Oxford Logic Guides, 36. Oxford University Press, New York, 1998, pp. 83–111.

[19] K. Kapulkin and P. L. Lumsdaine, *The simplicial model of univalent foundations (after Voevodsky)*. ***Journal of the European Mathematical Society (JEMS)***, vol. 23 (2021), no. 6, pp. 2071–2126. https://doi.org/10.4171/JEMS/1050.

[20] D. R. Licata and E. Finster, *Eilenberg-MacLane spaces in homotopy type theory*, ***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)***, ACM, New York, 2014, 66, 10. https://doi.org/10.1145/2603088.2603153.

[21] P. L. Lumsdaine, *Weak ω-categories from intensional type theory [reprint of MR2720193]*, ***Logical Methods in Computer Science Special issue: Selected Papers of the Conference "Typed Lambda Calculi and Applications 2009"*** (P-L. Curien, editor), ***vol.*** 3, 2010, 24, 19.

[22] 'E. Mangel and E. Rijke, *Delooping the sign homomorphism in univalent mathematics*, 2023. https://api.semanticscholar.org/CorpusID:256194575.

[23] P. Martin-Löf, *An intuitionistic theory of types: predicative part*, ***Logic Colloquium '73 (Bristol, 1973)*** (H. E. Rose and J. C. Shepherdson, editors), ***vol.*** 80, Studies in Logic and the Foundations of Mathematics, North-Holland, Amsterdam, 1975, pp. 73–118.

[24] A. Mörtberg, *Cubical methods in homotopy type theory and univalent foundations*. ***Mathematical Structures in Computer Science***, vol. 31 (2021), no. 10, pp. 1147–1184. https://doi.org/10.1017/s0960129521000311.

[25] A. Mörtberg and L. Pujet, *Cubical synthetic homotopy theory*, ***Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs***, CPP, Association for Computing Machinery, New Orleans, LA, 2020, pp. 158–171. https://doi.org/10.1145/3372885.3373825.

[26] D. J. Myers, ***Symmetry. Geometry. Modality***, PhD thesis, Johns Hopkins University, Baltimore, MD, 2022. https://jscholarship.library.jhu.edu/server/api/core/bitstreams/6d15a554-689d-4ba5-8d8d-b52a91de77a6/content.

[27] O Schreier, *Über die Erweiterung von Gruppen I*. ***Monatshefte für Mathematik und Physik***, vol. 34 (1926), no. 1, pp. 165–180. https://doi.org/10.1007/10.1007/BF01694897.

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

[29] P. Szekeres, ***A Course in Modern mathematical Physics: Groups, Hilbert Space and Differential Geometry***, Cambridge University Press, Cambridge, 2004, pp. xvi+600. https://doi.org/10.1017/CBO9780511607066.

[30] The univalent foundations program, *Homotopy Type Theory: Univalent Foundations of Mathematics*, Institute for Advanced Study, 2013.

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

[32] V. Voevodsky, *A very short note on homotopy λ-calculus*, 2006. https://www.math.ias.edu/~vladimir/Site3/Univalent_Foundations_files/Hlambda_short_current.pdf%7D.

TOPOS RESEARCH UK,
17 BEAUMONT STREET,
OXFORD,
OX1 2NA, UK
E-mail: davidjazmyers@gmail.com

CENTER FOR QUANTUM AND TOPOLOGICAL SYSTEMS,
DIVISION OF SCIENCE,
NEW YORK UNIVERSITY ABU DHABI,
SAADIYAT ISLAND, ABU DHABI,
UNITED ARAB EMIRATES
E-mail: zyad.yasser@nyu.edu

https://doi.org/10.1017/jsl.2024.78 Published online by Cambridge University Press