DELOOPING PRESENTED GROUPS IN HOMOTOPY TYPE THEORY

CAMIL CHAMPIN [a], SAMUEL MIMRAM [b], AND ÉMILE OLEON [b]

[a] École Normale Superieure de Lyon
e-mail address: camil.champin@ens-lyon.fr

[b] LIX, CNRS, École polytechnique, Institut Polytechnique de Paris, 91120 Palaiseau, France
e-mail address: samuel.mimram@polytechnique.edu, emile.oleon@polytechnique.edu

Abstract. Homotopy type theory is a logical setting based on Martin-Löf type theory in which one can perform geometric constructions and proofs in a synthetic way. Namely, types can be interpreted as spaces up to homotopy and proofs as homotopy invariant constructions. In this context, loop spaces of pointed connected groupoids provide a natural representation of groups, and any group can be obtained as the loop space of such a type, which is then called a *delooping* of the group. There are two main methods for constructing the delooping of an arbitrary group G. The first one consists in describing it as a pointed higher inductive type, whereas the second one consists in taking the connected component of the principal G-torsor in the type of sets equipped with an action of G. We show here that, when a presentation (or even a generating set) is known for the group, simpler variants of those constructions can be used to build deloopings. The resulting types are more amenable to computations and lead to simpler meta-theoretic reasoning. Finally, we develop a type theoretical notion of 2-polygraph, which allows manipulating higher inductive types such as the ones involved in the description of deloopings. These allow us to investigate in this context a construction for the Cayley graph of a generated group and show that it encodes the relations of the group, as well as a Cayley complex which encodes relations between relations. Many of the developments performed in the article have been formalized using the cubical version of the Agda proof assistant.

Introduction

Homotopy type theory was introduced around 2010 [39], based on Martin-Löf type theory [29]. It starts from the idea that types in logic should be interpreted not only as sets, as traditionally done in the semantics of logic, but rather as *spaces* considered up to homotopy. Namely, the identities between two elements of a type can be thought of as paths between points corresponding to the elements, identities on identities as homotopies between paths, and so on. Moreover, this correspondence can be made to work precisely, by postulating the *univalence axiom* [19], which states that identities between types coincide with equivalences. This opens the way to the implementation of geometric constructions in a synthetic manner, by performing operations on types, which will semantically correspond to the desired geometrical operations. In this setting, we are interested in providing ways to construct concise models of groups, allowing simple proofs, and making the meta-theoretic reasoning easier.

Preprint submitted to
Logical Methods in Computer Science

Delooping groups. Following a well-known construction due to Poincaré at the end of the 19th century [34], to any type A which is pointed, i.e. equipped with a distinguished element $\star$, we can associate its *fundamental group* $\pi_1 A$, defined as $\|\star = \star\|_0$, whose elements are the homotopy classes of paths from $\star$ to itself, with composition given by concatenation and identity by the constant path. Moreover, when the type A is a groupoid, in the sense that any two homotopies between paths are homotopic, this fundamental group coincides with the *loop space* $\Omega\, A$, which is $\star = \star$, i.e. defined similarly but without quotienting paths up to homotopy. Once this observation is made, it is natural to wonder whether every group G arises as the loop space of some groupoid. It turns out that this is the case: to every group one can associate a pointed connected groupoid type $\mathrm{B}\, G$, called its *delooping*, whose loop space is G. Moreover, there is essentially only one such type, thus justifying the notation.

Internal and external points of view. The delooping construction, which can be found in various places [3, 7], and will be recalled in the article, induces an equivalence between the type of groups and the type of pointed connected groupoids (Theorem 3). This thus provides us with two alternative descriptions of groups in homotopy type theory. The one as (loop spaces of) pointed connected groupoids can be thought of as an *internal* one, since the structure is deduced from the types without imposing further axioms; by opposition, the more traditional one as sets equipped with multiplication and unit operations is rather an *external* one (some also use the terminology *concrete* and *abstract* instead of internal and external [3]). We should also say here that pointed connected types (which are not necessarily groupoids) can be thought of as higher versions of groups, where the axioms only hold up to higher identities which are themselves coherent, and so on.

Two ways to construct deloopings. Two generic ways are currently known in order to construct the delooping $\mathrm{B}\, G$ of a group G, which we both refine in this article. The first one is the *torsor* construction which originates in algebraic topology [16] and can be adapted in homotopy type theory [3, 7, 41]. One can consider the type of *G-sets*, which are sets equipped with an action of G. Among those, there is a canonical one, called the *principal G-torsor* P_G, which arises from the action of the group G on itself by left multiplication. It can be shown that the loop space of the type of G-sets, pointed on the principal G-torsor P_G, is the group G. Moreover, if one restricts the type of G-sets to the connected component of the principal G-torsor, one obtains the type of *G-torsors*, which is a delooping of G.

The second one is a particular case of the definition of *Eilenberg-MacLane spaces* in homotopy type theory due to Finster and Licata [25]. It consists in constructing $\mathrm{B}\, G$ as a higher inductive type with one point, one loop for each element of G, one identity for each entry in the multiplication table of G, and then truncating the resulting type as a groupoid. One can imagine that the resulting space has the right loop space "by construction", although the formal proof is non-trivial.

Torsors for generated groups. In this article, we are interested in refining the above two constructions in order to provide ones which are "simpler" (in the sense that we have less constructors, or that the definition requires to introduce less material), when a presentation by generators and relation is known for the group. For the construction based on G-torsors, we show that a simpler definition can be achieved when a generating set X is known for G. Namely, we show that one can perform essentially the same construction, but replacing G-sets by what we call here X-sets (Theorem 15), where we only need to consider the action of the generators (as opposed to the whole group). As an illuminating example, consider the case $G = \mathbb{Z}$, whose delooping is known to be the circle $\mathrm{B}\,\mathbb{Z} = \mathrm{S}^1$. The type $\mathcal{U}^{\circlearrowleft}$ of all endomorphisms, on any type, contains, as a particular element, the successor function $s : \mathbb{Z} \to \mathbb{Z}$. Our results imply that the connected component of s in $\mathcal{U}^{\circlearrowleft}$ is a delooping of $\mathbb{Z}$. This description is arguably simpler than the one of $\mathbb{Z}$-torsors: indeed, morphisms of $\mathbb{Z}$-sets are required to preserve the action of every element of $\mathbb{Z}$, while morphisms in $\mathcal{U}^{\circlearrowleft}$ are only required to preserve the action of 1 (which corresponds to the successor). The above description is the one which is used in UniMath in order to define the circle S^1 [4]: the reason why they use it instead of more traditional one [39] is that they do not allow themselves to use higher inductive types because those are not entirely clear from a meta-theoretic point of view (there is no general definition, even though there are proposals [27], the semantics of type theory [19] has not been fully worked out in their presence, etc.). Our result thus give an abstract explanation about why this construction works and provides a generic way to easily define many more deloopings without resorting to higher inductive types, if one is not disposed to do so.

Higher inductive types for presented groups. For the second construction of the delooping, as a higher inductive type, we show here that we can construct $\mathrm{B}\,G$ as the higher inductive type generated by one point, one loop for each generator of the presentation (as opposed to every element of the group), one identity for each relation of the presentation and taking the groupoid truncation (Theorem 24). This has the advantage of resulting in types that are simpler to define, require handling fewer cases when reasoning with those by induction, and match the usual combinatorial descriptions of groups. It thus allows performing in a synthetical way proofs close to traditional ones in group theory, such as those based on Tietze transformations. Moreover, we claim that the traditional methods based on rewriting [2, 17] in order to compute invariants such as homology or coherence can be applied to those. Namely, a first important step in this direction was obtained by Kraus and von Raumer's adaptation of Squier's coherence theorem in homotopy type theory [23].

Polygraphs. In order to describe precisely the higher inductive types involved in the previous construction, and be able to reason internally on those, we recall the definition of 2-polygraphs, which originates in the study of rewriting and ω-categories [2] and was recently adapted to the setting homotopy type theory in [23]. We further develop their theory by introducing the notion of type generated and presented by a 2-polygraph, and show that the delooping $\mathrm{B}\,G$ constructed with higher inductive types precisely corresponds to a presented type. Finally, in order to show the applicability of traditional group-theoretic techniques, we develop a notion of Tietze transformation for 2-polygraphs and show that such transformations preserve the presented type (Theorem 29).

Cayley graphs. As another aspect of our study of generated groups in homotopy type theory, we provide here a pleasant abstract description of Cayley graphs, which is a well-known construction in group theory [12, 28]. We show that, given a group G with a set X of generators, the Cayley graph can be obtained as the kernel of the canonical map $\mathrm{B}\,X^* \to \mathrm{B}\,G$, where X^* is the free group on X (Theorem 35), and admits a nice description by a 2-polygraph. This establishes those graphs as a measure of the difference between deloopings and their approximations, and suggests higher dimensional versions of those: we also introduce Cayley complexes, which measure relations between relations (Theorem 40).

Formalization. Many of the results presented in this article have been formalized in the cubical variant of the Agda proof assistant [40] using the "standard library" which has been developed for it [1]. Our developments are publicly available [13], and we provide pointers to the formalized results when available.

Plan of the paper. We begin by briefly recalling the fundamental notions of homotopy type theory which will be used throughout the paper (Section 1), as well as the notion of delooping for a group (Section 2). We first present the construction of deloopings based on the torsor construction (Section 3) and show how it can be simplified when a generating set is known for the group (Section 4). We then present the other approach for defining deloopings of groups using higher inductive types, and explain how those can be simplified when a presentation is known for the group (Section 5). We introduce the structure of 2-polygraph and show that it allows encoding this notion of delooping of group, while supporting internal manipulations (Section 6). Finally, we investigate the construction of Cayley graphs and complexes in homotopy type theory (Section 7). We conclude, presenting possible extensions of this work (Section 8).

Acknowledgments. We would like to thank Dan Christensen as well as an anonymous reviewer for useful comments on early drafts of this article, which is an extended version of [14].

1. Homotopy type theory

In this section, we recall the main tools in homotopy type theory that will be used in the rest of the paper. A detailed presentation of the domain can be found in the reference books [39, 37].

Universe. We write $\mathcal{U}$ for the *universe*, i.e. the large type of all small types, which we suppose to be closed under dependent sums and products. We write 0, 1, 2 (and so on) for the initial type, the terminal type and the type of booleans. Given a type A and a type family $B : A \to \mathcal{U}$, we write $\Pi(x : A).B\,x$ or $(x : A) \to B\,x$ or $\Pi A.B$ for Π-types, and $A \to B$ for the case where B does not depend on A. Similarly, we write $\Sigma(x : A).B\,x$ or $\Sigma A.B$ for Σ-types, and $A \times B$ for the non-dependent version. The two projections from a Σ-type are respectively written $\pi : \Sigma A.B \to A$ and $\pi' : (x : \Sigma A.B) \to B(\pi x)$. Given type families $B : A \to \mathcal{U}$ and $B : A' \to \mathcal{U}$, a pair of maps $f : A \to A'$ and $g : (x : A) \to B\,x \to B'\,x$ canonically induces a map denoted $\Sigma f.g : \Sigma A.B \to \Sigma A'.B'$.

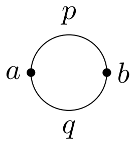
Paths. Given a type $A : \mathcal{U}$ and two elements $a, b : A$, we write $a =_A b$ (or simply $a = b$) for the type of *identities*, or *paths*, between a and b: its elements are proofs of equality between a and b. In particular, for any element $a : A$, the type $a = a$ contains the term refl_a witnessing for reflexivity of equality. We sometimes write $t \equiv u$ to indicate that t and u are equal by definition. The elimination principle of identities, aka *path induction* and often noted J, states that, given $a : A$, in order to show a property $P : (x : A) \to (a = x) \to \mathcal{U}$ for every $x : A$ and $p : a = x$ it is enough to show it in the case where $x \equiv a$ and $p \equiv \mathrm{refl}_a$. Given paths $p : x = y$ and $q : y = z$, we write $p \cdot q$ for their concatenation and $p^{-1} : y = x$ for the inverse of p.

Given a type A and a type family $B : A \to \mathcal{U}$, a path $p : x = y$ in A induces a function $B_p^{\to} : B(x) \to B(y)$ witnessing for the fact that equality is *substitutive*. As a special case, any path $p : A = B$ between two types $A, B : \mathcal{U}$ induces a function $p^{\rightarrow} : A \to B$, called the *transport* along p, as well as an inverse function $p^{\leftarrow} : B \to A$. Finally, given a function $f : A \to B$, any path $p : x = y$ in A induces a path $f^{=}(p) : f(x) = f(y)$ witnessing for the fact that equality is a *congruence*.

Pointed types. A *pointed* type consists of a type A together with a distinguished element, often written $\star_A$, or even $\star$, and sometimes left implicit. We write $\mathcal{U}_* \equiv \Sigma(A : \mathcal{U}).A$ for the type of pointed types. A *pointed map* $f : A \to B$ between pointed types is a map between the underlying types equipped with an identification $\star_f : f\star_A = \star_B$ and we write $A \to_* B$ for the type of pointed maps.

Higher inductive types. Many functional programming languages allow the definition of inductive types, which are freely generated by constructors. For instance, the type S^0 of booleans is generated by two elements (true and false). In the context of homotopy type theory, languages such as cubical Agda [40] feature a useful generalization of such types, called *higher inductive types*. They allow, in addition to traditional constructors for elements of the type, for constructors for equalities between elements of the type. For instance, the type corresponding to the circle S^1 can be defined as generated by two points a and b and two equalities $p, q : a = b$ between those points:

```
data S¹ : Type where
  a b : S¹
  p q : a ≡ b
```



(on the left is the definition of the higher inductive type in Agda, and on the right is pictured its geometric interpretation). Higher-dimensional spheres S^n, for n a natural number, can be defined in a similar way.

Univalence. A map $f : A \to B$ is an *equivalence* when it admits both a left and a right inverse, i.e. there are maps $g, g' : B \to A$ together with identities $g \circ f = \mathrm{id}_A$ and $f \circ g' = \mathrm{id}_B$. In particular, every isomorphism is an equivalence. We write $A \simeq B$ for the type of equivalences from A to B. The identity is clearly an equivalence and we thus have, by path induction, a canonical map

$$(A = B) \to (A \simeq B)$$

for every types A and B: the *univalence axiom* states that this map is itself an equivalence. In particular, every equivalence $A \simeq B$ induces a path $A = B$. It is known that univalence implies the *function extensionality* principle [39, Section 2.9]: given functions $f, g : A \to B$, if $f(x) = g(x)$ for any $x : A$ then $f = g$, and the expected generalization to dependent function types is also valid.

Homotopy levels. A type A is *contractible* when the type

$$\mathrm{isContr}(A) \quad \equiv \quad \Sigma(x : A).(y : A) \to (x = y)$$

is inhabited: this expresses the fact that we have a "contraction point" $x : A$, and a continuous family of paths from x to every other point y in A. A type A is a *proposition* (resp. a *set*, resp. a *groupoid*) when the type $(x = y)$ is contractible (resp. a proposition, resp. a set) for every $x, y : A$. Intuitively, a contractible type is a point (up to homotopy), a proposition is a point or is empty, a set is a collection of points and a groupoid is a space which bears no non-trivial 2-dimensional (or higher) structure. For instance, consider the following three types corresponding to spheres of respective dimensions −1, 0, 1 and 2:

The sphere S^{-1} (i.e. the empty type) is a proposition, the sphere S^0 (i.e. the booleans) is a set and the sphere S^1 (i.e. the circle) is a groupoid, and the sphere S^2 is not a groupoid (because there is a 2-dimensional "hole" in it). We write Set for the type of sets. Given a type A, we write $\mathrm{isSet}(A)$ (resp. $\mathrm{isGroupoid}(A)$) for the predicate indicating that A is a set (resp. a groupoid), i.e. we have

$$\mathrm{isSet}(A) \quad \equiv \quad (x, y : A) \to (p, q : x = y) \to \mathrm{isContr}(p = q)$$

and $\mathrm{isGroupoid}(A)$ is defined in a similar fashion.

Truncation. Given a type A, its *propositional truncation* turns it into a proposition in a universal way. It consists of a type $\|A\|_{-1}$, which is a proposition, equipped with a map $|-|_{-1} : A \to \|A\|_{-1}$ such that, for any proposition B, the map $(\|A\|_{-1} \to B) \to (A \to B)$ induced by precomposition by $|-|_{-1}$ is an equivalence:

$$\begin{tikzcd}
A \arrow[r] \arrow[d, "|-|_{-1}"'] & B \\
\|A\|_{-1} \arrow[ur, dotted] &
\end{tikzcd}$$

Intuitively, the type $\|A\|_{-1}$ behaves like A, except that we do not have access to its individual elements: the elimination principle for propositional truncation states that in order to construct an element of B from an element of $\|A\|_{-1}$, we can assume that we have an element of A only if B itself is a proposition. We say that a type is *merely* inhabited when its propositional truncation is inhabited. The *set truncation* $\|A\|_0$ of a type A is defined similarly, as the universal way of turning A into a set, and we write $|x|_0$ for the image of $x : A$ in the truncation; and we can similarly define the *groupoid truncation* $\|A\|_1$. A type A is *connected* when the type $\|A\|_0$ is a proposition or, equivalently, when the path type $x = y$ is merely inhabited for every elements x and y.

Fibers. Given a function $f : A \to B$ and an element $b : B$, the *fiber* of f at b is the type

$$\mathrm{fib}_f\, b \quad \equiv \quad \Sigma(a : A).(f\, a = b)$$

A function f is said to be *surjective* when the type $(b : B) \to \|\,\mathrm{fib}_f\, b\|_{-1}$ is inhabited, i.e. when every element of B merely admits a preimage. When B is pointed, we write $\ker A \equiv \mathrm{fib}_f \star$ for the fiber of f at $\star$, called its *kernel.* A sequence of composable arrows

$$F \longrightarrow A \xrightarrow{f} B$$

is a *fiber sequence* when F is the kernel of f and the map $F \to A$ is the first projection, see [39, Section 8.4].

Grothendieck duality. Any function $f : A \to B$ induces a type family $\mathrm{fib}_f : B \to \mathcal{U}$ by taking the fibers of f. Conversely, any type family $F : B \to \mathcal{U}$ induces a function, namely the first projection $\pi : \Sigma B.F \to B$. In fact, these two constructions can be shown to construct an equivalence, known as the *Grothendieck duality*, between types over B and type families indexed by B [39, Section 4.8]:

$$\Sigma(A : \mathcal{U}).(A \to B) \quad \simeq \quad (B \to \mathcal{U})$$

Switching between the two points of view is often very useful.

The flattening lemma. The *flattening lemma* allows for computing the total space of a fibration over a base type which is a colimit. We recall it here in the case of coequalizers and pushouts, and refer to [39, Section 6.12] for a more detailed presentation and proof. Suppose given a coequalizer

$$\begin{tikzcd} A \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & B \arrow[r, dotted, "h"] & C \end{tikzcd}$$

with $p : h \circ f = h \circ g$ witnessing for the coequalization, and a type family $P : C \to \mathcal{U}$. Then the diagram

$$\begin{tikzcd} \Sigma A.(P \circ h \circ f) \arrow[r, shift left, "\Sigma f.(\lambda\_.\,\mathrm{id})"] \arrow[r, shift right, "\Sigma g.e"'] & \Sigma B.(P \circ h) \arrow[r, dotted, "\Sigma h.(\lambda\_.\,\mathrm{id})"] & \Sigma C.P \end{tikzcd}$$

is a coequalizer, where the map

$$e : (a : A) \to P \circ h \circ f(a) \to P \circ h \circ g(a)$$

is induced by transport along p by $e\,a\,x \equiv P^{\rightarrow}_{p(a)}(x)$. Note that there is a slight asymmetry: we could have formulated a similar statement with $\Sigma A.(P \circ h \circ g)$ as left object.

The variant of the flattening lemma adapted to pushouts can be stated as follows. Consider a pushout square

$$\begin{tikzcd} X \arrow[r, "g"] \arrow[d, "f"'] & B \arrow[d, "j"] \\ A \arrow[r, "i"'] & A \sqcup_X B \arrow[ul, phantom, very near start, "\ulcorner"] \end{tikzcd}$$

with $p : i \circ f = j \circ g$ witnessing for its commutativity, together with a type family $P : A \sqcup_X B \to \mathcal{U}$. Then the following square of total spaces is also a pushout

$$\begin{tikzcd} \Sigma X.(P \circ i \circ f) \arrow[r, "\Sigma g.e"] \arrow[d, "\Sigma f.(\lambda\_.\,\mathrm{id})"'] & \Sigma B.(P \circ j) \arrow[d, "\Sigma j.(\lambda\_.\,\mathrm{id})"] \\ \Sigma A.(P \circ i) \arrow[r, "\Sigma i.(\lambda\_.\,\mathrm{id})"'] & \Sigma(A \sqcup_X B).P \end{tikzcd}$$

where

$$e : (x : X) \to P \circ i \circ f(x) \to P \circ j \circ g(x)$$

is the canonical morphism induced by p, i.e. $e(x) \equiv P^{\rightarrow}_{p(x)}$.

2. Delooping groups

The external point of view. A *group* consists of a set A, together with an operation $m : A \to A \to A$ (the *multiplication*), an element $e : A$ (the *unit*), and an operation $i : A \to A$ (the *inverse*) such that multiplication is associative, admits e as unit, and $i(x)$ is the two-sided inverse of any element $x : A$. We write Group for the type of all groups, and $G \to_{\mathrm{Grp}} H$ for the type of group morphisms between groups G and H:

$$\begin{aligned}\mathrm{Group} \quad \equiv \quad & \Sigma(A : \mathrm{Set}).(m : A \to A \to A) \times (e : A) \times (i : A \to A) \times \\ & \qquad ((x, y, z : A) \to m(m(x, y), z) = m(x, m(y, z))) \times \\ & \qquad ((x : A) \to (m(e, x) = x) \times (m(x, e) = x)) \times \\ & \qquad ((x : A) \to (m(i(x), x) = e) \times (m(x, i(x)) = e))\end{aligned}$$

In the following, we use the traditional notations for groups: given two elements $x, y : G$, we simply write xy instead of $m(x, y)$, 1 instead of e, and x^{-1} instead of $i(x)$.

Loop types. Given a pointed type A, its *loop space* $\Omega\, A$ is defined as the type of paths from $\star$ to itself, which are called the *loops* of A:

$$\Omega\, A \quad \equiv \quad (\star = \star)$$

This operation extends to a map

$$\Omega : \mathcal{U}_* \to \mathcal{U}_*$$

sending a pointed type A to the type $\Omega\, A$ pointed by $\mathrm{refl}_\star$. It is moreover functorial in the sense that any pointed map $f : A \to B$ induces a map $\Omega\, f : \Omega\, A \to \Omega\, B$ sending $p : \Omega\, A$ to the loop $\Omega\, fp \equiv \star_f^- \cdot f^= p \cdot \star_f$:

$$\begin{tikzcd} \star_B \arrow[r, equal, "\star_f^-"] & f\star_A \arrow[r, equal, "f^= p"] & f\star_A \arrow[r, equal, "\star_f"] & \star_B \end{tikzcd}$$

where the paths on the left and on the right are given by the fact that f is pointed, see [39, Definition 8.4.2]. This construction is compatible with composition and identities, in the sense that $\Omega(g \circ f) = \Omega\, g \circ \Omega\, f$ and $\Omega\, \mathrm{id} = \mathrm{id}$.

The internal point of view. By path induction one can construct, for every two paths $p : a = b$ and $q : b = c$, a path $p \cdot q : a = c$ called their *concatenation*. Similarly, every path $p : a = b$ admits an *inverse* path $p^- : b = a$. When A is a pointed groupoid, $\Omega\, A$ is a set, and these operations canonically equip this set with a structure of group [39, Section 2.1]. A pointed groupoid thus provides an *internal* notion of group in the sense that the group structure is induced by the inherent structure of the type (rather than being enforced by additional axioms). Moreover, this group structure only depends on the connected component of the canonical point, so that we might as well suppose that the space is connected.

A *delooping* of a group G is an internal representation of this group: it consists is a pointed connected groupoid $\mathrm{B}\, G$ together with an isomorphism of groups $\Omega\, \mathrm{B}\, G \cong G$. In such a situation, we often write $\mathrm{d}_G : \Omega\, \mathrm{B}\, G = G$ for the identity induced under univalence by the isomorphism. For instance, it is known that the circle is a delooping of $\mathbb{Z}$: indeed, S^1 is a connected groupoid, and its fundamental group is $\mathbb{Z}$ [39, Section 8.1].

Functoriality of delooping. We now recall that deloopings are unique when they exist, which follows from the fact that we can always deloop group morphisms. We have seen that for any pointed types A and B, the loop space function induces by functoriality a map

$$\Omega_{A,B} : (A \to_* B) \to (\Omega\, A \to \Omega\, B)$$

Furthermore, when A and B are pointed connected groupoids, this map is an equivalence. We provide a direct proof below, but this can also be recovered as the case $n = 0$ of [41, Corollary 12] (see also [3, Lemma 6.5.1]).

Proposition 1. *Given pointed connected groupoids A and B, the function $\Omega_{A,B}$ is an equivalence.*

Proof. By [39, Theorem 4.4.3], it is enough to show that the fiber of $\Omega_{A,B}$ at any morphism $g : \Omega\, A \to_{\mathrm{Grp}} \Omega\, B$ is contractible. By definition, the fiber of $\Omega_{A,B}$ at g is

$$\mathrm{fib}_{\Omega_{A,B}}\, g \quad = \quad \Sigma(f : A \to B).\Sigma(\star_f : f\star_A = \star_B).(\Omega\, f = g)$$

This type can be shown to be equivalent to the type

$$\Sigma(f : A \to B).\Pi(a : A).C(a, f(a)) \tag{2.1}$$

which means that maps $f : A \to B$ which are deloopable are those such that for every $a : A$ the image satisfies $C(a, f(a))$. Above, for $a : A$ and $b : B$, the type $C(a, b)$ is defined as

$$\Sigma(\pi : (a = \star_A) \to (b = \star_B)).D(a, \pi)$$

with $D(a, \pi)$ defined as

$$\Pi(p : a = \star_A).\Pi(q : \star_A = \star_A).\pi(p \cdot q) = \pi(p) \cdot g(q) \tag{2.2}$$

see [41, Lemma 9]. Namely, we have

$$\begin{aligned}
\mathrm{fib}_{\Omega_{A,B}}\, g &= \Sigma(f : A \to B).\Sigma(\star_f : f\star_A = \star_B).\ \Omega_{A,B}\, f = g \\
&\hspace{20em}\text{by definition} \\
&= \Sigma(f : A \to B).\Sigma(\star_f : f\star_A = \star_B).(q : \star_A = \star_A) \to \Omega_{A,B}\, f(q) = g(q) \\
&\hspace{16em}\text{by function extensionality} \\
&= \Sigma(f : A \to B).\Sigma(\star_f : f\star_A = \star_B).(q : \star_A = \star_A) \to \star_f^- \cdot f^= q \cdot \star_f = g(q) \\
&\hspace{18em}\text{by definition of } \Omega_{A,B} \\
&= \Sigma(f : A \to B).\Sigma(\star_f : f\star_A = \star_B).(q : \star_A = \star_A) \to f^= q \cdot \star_f = \star_f \cdot g(q) \\
&\hspace{14em}\text{by identities on path composition} \\
&= \Sigma(f : A \to B).\Sigma(\pi : (\Sigma(a : A).(a = \star_A)) \to (f(a) = \star_B)). \\
&\qquad \Pi(q : \star_A = \star_A).\pi\, \star_a\, q = \pi\, \star_A\, \mathrm{refl} \cdot g(q) \\
&\hspace{15em}\text{by contractibility of singletons} \\
&= \Sigma(f : A \to B).\Sigma(\pi : (a : A) \to (a = \star_A) \to (f(a) = \star_B)). \\
&\qquad \Pi(q : \star_A = \star_A).\pi\, \star_a\, q = \pi\, \star_A\, \mathrm{refl} \cdot g(q) \\
&\hspace{20em}\text{by curryfication} \\
&= \Sigma(f : A \to B).\Sigma(\pi : (a : A) \to (a = \star_A) \to (f(a) = \star_B)). \\
&\qquad \Pi(q : \star_A = \star_A).\pi\, \star_a\, (\mathrm{refl} \cdot q) = \pi\, \star_A\, \mathrm{refl} \cdot g(q)
\end{aligned}$$

$$\begin{aligned}
&\hspace{20em}\text{by unitality of refl}\\
&= \Sigma(f : A \to B).\Sigma(\pi : (a : A) \to (a = \star_A) \to (f(a) = \star_B)).\\
&\qquad \Pi((a,p) : \Sigma(a : A).a = \star_A).\Pi(q : \star_A = \star_A).\pi\, a\, (p \cdot q) = \pi\, a\, (p) \cdot g(q)\\
&\hspace{20em}\text{by contractibility of singletons}\\
&= \Sigma(f : A \to B).\Sigma(\pi : (a : A) \to (a = \star_A) \to (f(a) = \star_B)).\\
&\qquad \Pi(a : A).\Pi(p : a = \star_A).\Pi(q : \star_A = \star_A).\pi\, a\, (p \cdot q) = \pi\, a\, (p) \cdot g(q)\\
&\hspace{20em}\text{by curryfication}\\
&= \Sigma(f : A \to B).\Pi(a : A).\Sigma(\pi : (a = \star_A) \to (f(a) = \star_B)).\Pi(p : a = \star_A).\\
&\qquad \Pi(q : \star_A = \star_A).\pi(p \cdot q) = \pi(p) \cdot g(q)\\
&\hspace{20em}\text{by type theoretic choice}\\
&= \Sigma(f : A \to B).\Pi(a : A).C(a, f(a))
\end{aligned}$$

Above, the "type theoretic choice" is the equivalence for any types $A, B : \mathcal{U}$ and type family $P : A \to B \to \mathcal{U}$,

$$\Pi(x : A).\Sigma(y : B).P\, x\, y \qquad \simeq \qquad \Sigma(f : A \to B).\Pi(x : A).P\, x\, (f\, x)$$

see [39, Section 1.6], and the "contractibility of singletons" is the fact that, for any type A, the type $\Sigma(x : A).(x = \star)$ is contractible, see [39, Lemma 3.11.8]. By type theoretic choice again, we deduce that $\mathrm{fib}_{\Omega_{A,B}}\, g$ is also equivalent to

$$\Pi(a : A).\Sigma(b : B).C(a, b) \tag{2.3}$$

We show below that we have, for any $b : B$,

$$C(\star_A, b) \quad = \quad (b = \star_A) \tag{2.4}$$

From there we deduce that $\Sigma(b : B).C(\star_A, b)$ is equivalent to $\Sigma(b : B).(b = \star_A)$ and is thus contractible (by contractibility of singletons). By path induction, we thus have

$$\Pi(a : A).(a = \star_A) \to \mathrm{isContr}(\Sigma(b : B).C(a, b))$$

and thus

$$\Pi(a : A).\|a = \star_A\|_{-1} \to \mathrm{isContr}(\Sigma(b : B).C(a, b))$$

because being contractible is a proposition. Since A is connected, we deduce that the type $\Sigma(b : B).C(a, b)$ is contractible for every $a : A$, and the type (2.3) is thus also contractible. We have thus also $\mathrm{fib}_{\Omega_{A,B}}\, g$, which is what we wanted to show.

We are left with showing (2.4). It can be shown that, for any suitably typed function π, the type $D(\star_A, \pi)$ is equivalent to

$$\Pi(q : \star_A = \star_A).\pi(q) = \pi(\mathrm{refl}) \cdot g(q)$$

Namely, the former implies the later as a particular case and, conversely, supposing the second one, we have for $p : \star_A = \star_A$ and $q : \star_A = \star_A$,

$$\pi(p \cdot q) = \pi(\mathrm{refl}) \cdot g(p \cdot q) = \pi(\mathrm{refl}) \cdot g(p) \cdot g(q) = \pi(p) \cdot g(q)$$

because g preserves composition. From there follows easily (2.4), i.e. that

$$\Sigma(\pi : (\star_A = \star_A) \to (b = \star_B)).\pi(q) = \pi(\mathrm{refl}) \cdot g(q)$$

is equivalent to $b = \star_B$, since, in the above type, the second component expresses that the function π in the first component is uniquely determined by $\pi(\mathrm{refl})$, which is an element of $b = \star_B$. □

Given a function $g : \Omega\, A \to \Omega\, B$, the previous proposition ensures that there is a unique function $f : A \to_* B$ such that $\Omega\, f = g$. This function is noted $\mathrm{B}\, g$ and called the *delooping* of g. As another immediate consequence of Proposition 1, deloopings are unique:

Proposition 2. *Given two deloopings* $\mathrm{B}\, G$ *and* $\mathrm{B}'\, G$ *of a group* G*, we have* $\mathrm{B}\, G = \mathrm{B}'\, G$.

Proof. Given two deloopings $\mathrm{B}\, G$ and $\mathrm{B}'\, G$, we have an equivalence $f : \Omega\,\mathrm{B}\, G \simeq \Omega\,\mathrm{B}'\, G : g$, which lifts as two morphisms $\mathrm{B}\, f : \mathrm{B}\, G \to \mathrm{B}'\, G$ and $\mathrm{B}\, g : \mathrm{B}'\, G \to \mathrm{B}\, G$. By the uniqueness of Proposition 1, delooping of morphisms preserves composition and identities, and f and g thus form an equivalence. ☐

Given a group morphism $f : G \to H$ such that both G and H admit deloopings (and we will see that this is actually always the case by Theorems 13 and 24), the *delooping* of f is the morphism

$$\mathrm{B}\, f : \mathrm{B}\, G \to \mathrm{B}\, H$$

associated, by Proposition 1, to the morphism $\mathrm{d}^{\leftarrow}_{H} \circ f \circ \mathrm{d}^{\rightarrow}_{G} : \Omega\,\mathrm{B}\, G \to \Omega\,\mathrm{B}\, H$. By Proposition 1, this operation is functorial in the sense that that it preserves identities and composition.

Equivalence between the two points of view. Although this is not central in this article, we shall mention here the fundamental equivalence provided by the above constructions; details can be found in [3]. We write IntGroup for the type of internal groups, i.e. pointed connected groupoids.

Theorem 3. *The maps* $\Omega : \mathrm{IntGroup} \to \mathrm{Group}$ *and* $\mathrm{B} : \mathrm{Group} \to \mathrm{IntGroup}$ *form an equivalence of types.*

Proof. Given a group G, we have $\Omega\,\mathrm{B}\, G = G$ by definition of $\mathrm{B}\, G$. Given an internal group A, we have $\mathrm{B}\,\Omega\, A \simeq A$ by Proposition 2. ☐

The above theorem thus states looping and delooping operators allow us to go back and forth between the external and the internal point of view of group theory in homotopy type theory.

Internal group actions. In a similar way that the traditional notion of group admits an internal reformulation (Section 2), the notion of action also admits an internal counterpart which can be defined as follows. Given a group G, an *internal* action of G on a set A is a function

$$\alpha : \mathrm{B}\, G \to \mathrm{Set}$$

such that $\alpha(\star) = A$. Since Set is a groupoid [39, Theorem 7.1.11], by Theorem 3, we have equivalences of types

$$(\mathrm{B}\, G \to \mathrm{Set}) \quad\simeq\quad (\Omega\,\mathrm{B}\, G \to \Omega(\mathrm{Set}, A)) \quad\simeq\quad (G \to \mathrm{Aut}\, A)$$

which shows that internal group actions correspond to external ones: the delooping operator internalizes an external group action, and the looping operator externalizes an internal group action.

3. Delooping with torsors

In this section, we recall a classical approach to constructing deloopings of groups by using G-torsors, which originates in classical constructions of algebraic topology [16]. Most of the material of the section is already known, which is why the proofs are not as detailed. A more in-depth presentation can be found in recent works such as [3, 7].

Group actions. Given a group G and a set A, an *action* of G on A is a group morphism from G to $A \simeq A$, that is a map $\alpha : G \to (A \simeq A)$ such that

$$\alpha(xy) = \alpha(x) \circ \alpha(y) \qquad\qquad \alpha(1) = \mathrm{id}_A \tag{3.1}$$

for all $x, y : G$.

A *G-set* is a set equipped with an action of G, and we write Set_G for the type of G-sets. We often simply denote a G-set by the associated action α and write $\mathrm{dom}(\alpha)$ for the set on which G acts.

Lemma 4 (`GSetProperties.isGroupoidGSet`)**.** *The type* Set_G *is a groupoid.*

Proof. The type of sets is a groupoid [39, Theorem 7.1.11]. Given a set A, the type of functions $A \to A$ is a set [39, Theorem 7.1.9] and thus a groupoid. Finally, the axioms (3.1) of actions are propositions (because A is a set) and thus groupoids. We conclude since groupoids are closed under Σ-types [39, Theorem 7.1.8]. □

Given G-sets α and β, a *morphism* between them consists of a function $f : \mathrm{dom}\,\alpha \to \mathrm{dom}\,\beta$ which preserves the group action, in the sense that for every $x : G$ and $a : \mathrm{dom}\,\alpha$, we have

$$\beta(x)(f(a)) = f(\alpha(x)(a)) \tag{3.2}$$

A morphism which is also an equivalence is called an *isomorphism* and we write $\alpha \simeq^{\mathrm{Set}_G} \beta$ for the type of isomorphisms between α and β. We write $\mathrm{Aut}(\alpha)$ for the type of automorphisms $\alpha \simeq^{\mathrm{Set}_G} \alpha$, which is a group under composition. The equalities between G-sets can be conveniently characterized as follows.

Proposition 5 (`GSetProperties.GSet≡Decomp`)**.** *Given two G-sets α and β, an equality between them consists of an equality $p : \mathrm{dom}\,\alpha = \mathrm{dom}\,\beta$ such that the function induced by transport along p, namely $p^{\to} : \mathrm{dom}\,\alpha \to \mathrm{dom}\,\beta$, is a morphism of G-sets.*

Proof. The characterization of equalities between Σ-types [39, Theorem 2.7.2] entails that an equality between $(\mathrm{dom}\,\alpha, \alpha)$ and $(\mathrm{dom}\,\beta, \beta)$ is a pair equalities

$$p : \mathrm{dom}\,\alpha = \mathrm{dom}\,\beta \qquad\qquad q : p^{\to}(\alpha) = \beta$$

(we can forget about the equality between the components expressing the properties required for group actions since those are propositions). By [39, Lemma 2.9.6] and function extensionality, we finally have that the type of q is equivalent to the type $\beta(x) \circ p^{\to} = p^{\to} \circ \alpha(x)$. □

It easily follows from this proposition that any equality between G-sets induces an isomorphism of G-sets, as customary for equalities between algebraic structures [39, Section 2.14]. In fact, this map from equalities to isomorphisms can itself be shown to be an equivalence:

Proposition 6 (`GSetProperties.GSetPath`)**.** *Given G-sets α and β, the canonical function*

$$(\alpha = \beta) \to (\alpha \simeq^{\mathrm{Set}_G} \beta)$$

is an equivalence. Moreover, given a G-set α, the induced equivalence

$$(\alpha = \alpha) \simeq (\alpha \simeq^{\mathrm{Set}_G} \alpha)$$

is compatible with the canonical group structures on both types.

Proof. This is actually an instance of a more general correspondence between equalities and isomorphisms of algebraic structures, which is known under the name of *structure identity principle*, see [15] and [39, Section 9.8], and can be understood as a generalisation of univalence for types having an algebraic structure. □

Connected components. In the following, in order to define torsors, we will need to use a type theoretic counterpart for the notion of *connected component* of a pointed type A: this is the type of points which are merely connected to the distinguished point of A. This type is noted $\operatorname{Comp} A$ (or $\operatorname{Comp}(A, \star)$ when we want to specify the distinguished element $\star$). Formally,

$$\operatorname{Comp} A \quad \equiv \quad \Sigma(x : A).\|\star = x\|_{-1}$$

This type is canonically pointed by $(\star, |\operatorname{refl}|_{-1})$. This construction deserves its name because it produces a connected space, whose geometry is the same as the original space around the distinguished point, as shown in the following two lemmas.

Lemma 7 (`Comp.isConnectedComp`). *The type* $\operatorname{Comp} A$ *is connected.*

Proof. It can be shown that a type X is connected precisely when both $\|X\|_{-1}$ and $(x, y : X) \to \|x = y\|_{-1}$ are inhabited, i.e. when X merely has a point and any two points are merely equal [39, Exercise 7.6]. In our case, the type $\operatorname{Comp} A$ is pointed and thus $\|\operatorname{Comp} A\|_{-1}$ holds. Moreover, suppose that there are two points (x, p) and (y, q) in $\operatorname{Comp} A$ with $x, y : A$, $p : \|\star = x\|_{-1}$ and $q : \|\star = y\|_{-1}$. Our goal is to show that $\|(x, p) = (y, q)\|_{-1}$ holds, which is a proposition, so by elimination of propositional truncation, we can therefore assume that p (resp. q) has type $\star = x$ (resp. $\star = y$). Hence, we can construct a path $p^- \cdot q$ of type $x = y$, and therefore $(x, p) = (y, q)$ because the second components belong to a proposition by propositional truncation. We conclude that $\|(x, p) = (y, q)\|_{-1}$ and finally that $\operatorname{Comp} A$ is connected. □

Lemma 8 (`Comp.loopCompIsLoop`). *We have* $\Omega \operatorname{Comp} A = \Omega A$.

Proof. We begin by showing that the type

$$\Sigma((x, t) : \operatorname{Comp} A).(\star = x) \tag{3.3}$$

is contractible. In order to do so, observe that we have the following equivalence of types:

$$\begin{aligned}
\Sigma((x, t) : \operatorname{Comp} A).(\star = x) &\simeq \Sigma((x, t) : \Sigma(x : A).\|\star = x\|_{-1}).(\star = x) \\
&\simeq \Sigma(x : A).(\|\star = x\|_{-1}) \times (\star = x) \\
&\simeq \Sigma((x, p) : \Sigma(x : A).(\star = x)).\|\star = x\|_{-1}
\end{aligned}$$

using classical associativity and commutativity properties of Σ-types. Moreover, the type $\Sigma(x : A).(\star = x)$ is contractible [39, Lemma 3.11.8], therefore the whole type on the last line is a proposition (as a sum of propositions over a proposition), and therefore also the original type (3.3). We write $\star'$ for the element $(\star, \|\operatorname{refl}_\star\|_{-1})$ of $\operatorname{Comp} A$. The type (3.3) is pointed by the canonical element $(\star', \operatorname{refl})$ and thus contractible as a pointed proposition.

We have a morphism

$$\begin{array}{rccl} F : ((x,t) : \operatorname{Comp} A) & \to (\star' = (x,t)) & \to & (\star = x) \\ (x,t) & p & \mapsto & \pi^{=}(p) \end{array}$$

sending a path p to the path obtained by applying the first projection. It canonically induces a morphism

$$\begin{gathered} \Sigma((x,t) : \operatorname{Comp} A).(\star' = (x,t)) \to \Sigma((x,t) : \operatorname{Comp} A).(\star = x) \\ ((x,t),p) \mapsto ((x,t),\pi^{=}(p)) \end{gathered}$$

between the corresponding total spaces. Since the left member is contractible (by [39, Lemma 3.11.8] again) and the right member is also contractible (as shown above), this is an equivalence. By [39, Theorem 4.7.7], for every $x : \operatorname{Comp} A$, the fiber morphism $F x$ is also an equivalence. In particular, with x being $\star'$, we obtain $\Omega \operatorname{Comp} A \simeq \Omega A$ (as a type) and we can conclude by univalence. Note that the equivalence preserves the group structure so that the equality also holds in groups. □

As a direct corollary of the two above lemmas, we have:

Proposition 9. *Given a pointed groupoid A, $\operatorname{Comp} A$ is a delooping of ΩA.*

Remark 10. Some people write $\operatorname{Aut} A$ for ΩA and the above proposition states that we have $\mathrm{B}\operatorname{Aut} A = \operatorname{Comp} A$. For this reason, the (confusing) notation $\mathrm{BAut}\, A$ is also found in the literature for $\operatorname{Comp} A$.

Torsors. For any group G, there is a canonical G-set called the *principal G-torsor* and noted P_G, corresponding to the action of G on itself by left multiplication. Moreover, its group of automorphisms is precisely the group G:

Proposition 11 (`Deloopings.PGloops`)**.** *Given a group G, we have an equality of groups*

$$(\mathrm{P}_G \simeq^{\mathrm{Set}_G} \mathrm{P}_G) = G$$

Proof. The two functions

$$\begin{array}{rlcrl} \phi : \operatorname{Aut} \mathrm{P}_G \to G & & \qquad & \psi : G \to \operatorname{Aut} \mathrm{P}_G & \\ f \mapsto f(1) & & & x \mapsto y \mapsto yx & \end{array}$$

are group morphisms. Namely, given $f, g : \operatorname{Aut} \mathrm{P}_G$, we have

$$\phi(g \circ f) = g \circ f(1) = g(f(1)1) = f(1)g(1) = \phi(f)\phi(g) \qquad \phi(\mathrm{id}) = \mathrm{id}(1) = 1$$

and given $x, y : G$, we have for every $z : G$,

$$\psi(xy)(z) = z(xy) = (zx)y = \psi(y) \circ \psi(x)(z) \qquad \psi(1)(x) = x1 = \mathrm{id}(x)$$

Moreover, they are mutually inverse. Namely, given $f : \operatorname{Aut} \mathrm{P}_G$ and $x : G$, we have

$$\psi \circ \phi(f)(x) = xf(1) = f(x1) = f(x) \qquad \phi \circ \psi(x) = 1x = x$$

We thus have $\operatorname{Aut} \mathrm{P}_G \simeq G$ and we conclude by univalence. □

The type Set_G is thus “almost” a delooping of G. Namely, it is a groupoid (Lemma 4), which is pointed by P_G and satisfies $\Omega\, \mathrm{Set}_G = G$ by Propositions 6 and 11. It only lacks being connected, which is easily addressed by restricting to the connected component.

Definition 12. The type of *G-torsors* is the connected component of P_G in Set_G.

Theorem 13 (`Deloopings.torsorDeloops`)**.** *The type of G-torsors is a delooping of G.*

Note that the torsor construction only gives a delooping in a larger universe than the original group unless one makes additional assumptions such as the *replacement axiom* [37, Axiom 18.1.8].

4. Generated torsors

Free groups. Given a set X, we write X^* for the *free group* over X [39, Theorem 6.11.6]. There is an inclusion function $\iota : X \to X^*$ which, by precomposition, induces an equivalence between morphisms of groups $X^* \to_{\mathrm{Grp}} G$ and functions $X \to G$:

$$\begin{tikzcd}
X \arrow[r, "f"] \arrow[d, "\iota"'] & G \\
X^* \arrow[ur, dotted, "\tilde{f}"'] &
\end{tikzcd}$$

We write $f^* : X^* \to G$ for the group morphism thus induced by a function $f : X \to G$. The elements of X^* can be described as formal composites $a_1 \ldots a_n$ where each a_i is an element of X or a formal inverse of an element of X (such that an element with an adjacent formal inverse cancel out).

Generated groups. Fix a group G. Given a set X and a map $\gamma : X \to G$, we say that X *generates* G (with respect to γ) when $\gamma^* : X^* \to G$ is surjective. From now on, we suppose that we are in such a situation. We now provide a variant for the construction of a delooping of G by G-torsors described in the previous section, taking advantage of the additional data of a generating set in order to obtain smaller and simpler constructions. Note that here, contrarily to Section 5, we only need a set of generators, not a full presentation.

Actions of sets. Given a type A, we write $\operatorname{End} A$ for its type of *endomorphisms*, i.e. maps $A \to A$. An *action* of the set X on a set A is a morphism $X \to \operatorname{End} A$, i.e. a family of endomorphisms of A indexed by X. We write Set_X for the type

$$\mathrm{Set}_X \quad \equiv \quad \Sigma(A : \mathrm{Set}).(X \to \operatorname{End} A)$$

of actions of X. An element α of this type consists in a set $\operatorname{dom} \alpha$ with a function $\alpha : X \to \operatorname{End}(\operatorname{dom} \alpha)$ and is called an *X-set*. A *morphism* between X-sets α and β is a function $f : \operatorname{dom} \alpha \to \operatorname{dom} \beta$ satisfying (3.2) for every $x : X$. The identities between X-sets can be characterized in a similar way as for G-sets, see Proposition 5, and Proposition 6 also extends in the expected way.

Precomposition by γ induces a function $U : \mathrm{Set}_G \to \mathrm{Set}_X$ which can be thought of as a forgetful functor from G-sets to X-sets. Note that U depends on γ but we leave it implicit for concision.

Applications of the generated delooping. We have seen in the previous section that the connected component of the principal G-torsor P_G in G-sets is a delooping of G. Our aim in this section is to show here that this construction can be simplified by taking the connected component of the restriction of P_G to X-sets.

Before proving this theorem, which is formally stated as Theorem 15 below, we shall first illustrate its use on a concrete example. Consider $\mathbb{Z}_n$, the cyclic group with n elements. We write $s : \mathbb{Z}_n \to \mathbb{Z}_n$ for the successor (modulo n) function, which is an isomorphism. By Theorem 13, we know that the type

$$\Sigma(A : \mathrm{Set}_{\mathbb{Z}_n}).\| \mathrm{P}_{\mathbb{Z}_n} = A\|_{-1}$$

of $\mathbb{Z}_n$-torsors is a model of $\mathrm{B}\,\mathbb{Z}_n$. This type is the connected component of the principal $\mathbb{Z}_n$-torsor $\mathrm{P}_{\mathbb{Z}_n}$ in the universe $\mathrm{Set}_{\mathbb{Z}_n}$ of sets with an action of $\mathbb{Z}_n$, i.e. sets A equipped with a morphism $\alpha : \mathbb{Z}_n \to \mathrm{Aut}\,A$. Such a set A thus comes with one automorphism $\alpha(k)$ for every element $k : \mathbb{Z}_n$, therefore k automorphisms in this case. However, most of them are superfluous: 1 generates all the elements of $\mathbb{Z}_n$ by addition, so $\alpha(1)$ generates all the $\alpha(k)$ by composition because $\alpha(k) = \alpha(1)^k$. The useful data of a $\mathbb{Z}_n$-set thus boils down to a set A together with one automorphism $\alpha : \mathrm{Aut}\,A$ such that $\alpha^n = \mathrm{id}_A$.

Indeed, writing $\mathrm{Set}^{\circlearrowleft} \equiv \Sigma(A : \mathrm{Set}).\,\mathrm{End}\,A$ for the type of all *endomorphisms* (on any set), our theorem will imply that the type

$$\Sigma((A, f) : \mathrm{Set}^{\circlearrowleft}).\|(\mathbb{Z}_n, s) = (A, f)\|_{-1} \tag{4.1}$$

(the connected component of the successor modulo n in the universe of set endomorphisms) is still a delooping of $\mathbb{Z}_n$. Note that we didn't assume that f is an isomorphism nor that it should verify $f^n = \mathrm{id}$. This is because both properties follow from the fact that f is in the connected component of the successor (which satisfies those properties). Similarly, we do not need to explicitly assume that the domain of the endomorphism is a set.

Our theorem thus allows to define, in a relatively simple way, types corresponding to deloopings of groups. As recalled in the introduction, this is particularly useful when one is not disposed to use higher inductive types (e.g. because their definition, implementation and semantics are not entirely mature). This is in fact the reason why this approach was used in UnitMath to define the circle [4], and we provide a generic way to similarly define other types. We expect that it can be used to reason about groups and compute invariants such as their cohomology [11, 8, 6]. On a side note, one might be worried by the fact that we are "biased" (by using a particular set of generators), which allows us to be more concise but might make more difficult generic proofs compared to G-torsors: we expect that this is not the case because in order to define the group G itself, one usually needs to resort to a presentation, and thus is also biased in some sense...

The generated delooping. In the following, we write P_X for $U\,\mathrm{P}_G$.

Proposition 14 (`XSetProperties.theorem`). *We have a group equivalence* $\Omega\,\mathrm{P}_G \simeq \Omega\,\mathrm{P}_X$.

Proof. From Proposition 5, an element of $\Omega\,P_G$ consists of an equality $p : G = G$ in $\mathcal{U}$ such that

$$\mathrm{P}_G(x) \circ p^{\to} = p^{\to} \circ \mathrm{P}_G(x)$$

for every $x : G$. By function extensionality and the definition of the action P_G, this is equivalent to requiring, for every $g, z : G$ that

$$g(p^{\to}(z)) = p^{\to}(gz) \tag{4.2}$$

Note that the above equality is between elements of G, which is a set, and is thus a proposition. Similarly, an element of $\Omega\,\mathrm{P}_X$ consists of an equality $p : G = G$ in $\mathcal{U}$ satisfying

$$\gamma(x)(p^{\rightarrow}(z)) = p^{\rightarrow}(\gamma(x)z) \tag{4.3}$$

for every $x : X$ and $z : G$.

Clearly, any equality $p : G = G$ in $\Omega\,\mathrm{P}_G$ also belongs to $\Omega\,\mathrm{P}_X$ since the condition (4.3) is a particular case of (4.2). We thus have a function $\phi : \Omega\,\mathrm{P}_G \to \Omega\,\mathrm{P}_X$. Conversely, consider an element $p : G = G$ of $\Omega\,\mathrm{P}_X$, i.e. satisfying (4.3) for every $x : X$ and $z : G$. Our aim is to show that it belongs to $\Omega\,\mathrm{P}_G$. Given $g, z : G$, we thus want to show that (4.2) holds. Since γ^* is surjective, because X generates G, we know that there merely exists an element u of X^* such that $\gamma^*(u) = x$. Since (4.2) is a proposition, by the elimination principle of propositional truncation, we can actually suppose given such a u, and we have

$$\begin{aligned}
x(p^{\rightarrow}(y)) &= \gamma^*(u)(p^{\rightarrow}(y)) && \text{since } \gamma^*(u) = x \\
&= p^{\rightarrow}(\gamma^*(u)y) && \text{by repeated application of (4.3)} \\
&= p^{\rightarrow}(xy) && \text{since } \gamma^*(u) = x.
\end{aligned}$$

The second equality essentially corresponds to the commutation of the following diagram, where $u \equiv x_1 x_2 \ldots x_n$ with $x_i : X$:

$$\begin{tikzcd}
G \arrow[r, "\gamma(x_1)"] \arrow[d, "p^{\rightarrow}"'] & G \arrow[r, "\gamma(x_2)"] \arrow[d, "p^{\rightarrow}"'] & G \arrow[r] \arrow[d, "p^{\rightarrow}"'] & \ldots \arrow[r] & G \arrow[r, "\gamma(x_n)"] \arrow[d, "p^{\rightarrow}"] & G \arrow[d, "p^{\rightarrow}"] \\
G \arrow[r, "\gamma(x_1)"'] & G \arrow[r, "\gamma(x_2)"'] & G \arrow[r] & \ldots \arrow[r] & G \arrow[r, "\gamma(x_n)"'] & G
\end{tikzcd}$$

This thus induces a function $\psi : \Omega\,\mathrm{P}_X \to \Omega\,\mathrm{P}_G$. The functions ϕ and ψ clearly preserve the group structure (given by concatenation of paths) and are mutually inverse of each other, hence we have the equivalence we wanted. □

Theorem 15. *The type* $\mathrm{Comp}\,\mathrm{P}_X$ *is a delooping of* G.

Proof. Since taking the connected component preserves loops spaces (Proposition 9), we have that $\Omega\,\mathrm{Comp}\,\mathrm{P}_X$ is equal to $\Omega\,\mathrm{P}_X$, which in turn is equal to $\Omega\,\mathrm{P}_G$ by Proposition 14, and thus to G by Theorem 13. □

The delooping of G constructed in previous theorem is the component of P_X in X-sets:

$$\Sigma(A : \mathcal{U}).\Sigma(S : \mathrm{isSet}\,A).\Sigma(f : X \to \mathrm{End}\,A).\|\,\mathrm{P}_X = (A, S, f)\|_{-1}$$

Since for any type A, the type $\mathrm{isSet}\,A$ is a proposition [39, Theorem 7.1.10], and the underlying type of P_X is a set, the underlying type of any X-set in the connected component of P_X will also be a set. As a consequence, the above type can slightly be simplified, by dropping the requirement that A should be a set:

Proposition 16. *The type* $\Sigma(A : \mathcal{U}).\Sigma(f : X \to \mathrm{End}\,A).\|\,\mathrm{P}_X = (A, f)\|_{-1}$ *is a delooping of* G.

For instance, the delooping (4.1) of $\mathbb{Z}_n$ can slightly be simplified as

$$\Sigma((A, f) : \mathcal{U}^{\circlearrowleft}).\|(\mathbb{Z}_n, s) = (A, f)\|_{-1}$$

where $\mathcal{U}^{\circlearrowleft} \equiv \Sigma(A : \mathcal{U}).\,\mathrm{End}\,A$ is the type of all endomorphisms of the universe.

Example 17. Theorem 15 applies to every group for which a generating set is known (and, of course, the smaller the better). For instance, given a natural number n, the *dihedral group* D_n is the group of symmetries of a regular polygon with n sides. It has $2n$ elements and is generated by two elements s (axial symmetry) and r (rotation by an angle of $2\pi/n$). Hence the connected component of the symmetry and the rotation in the type of pairs of set endomorphisms, i.e.

$$\Sigma(A : \mathrm{Set}).\Sigma(f, g : \operatorname{End} A \times \operatorname{End} A).\|(D_n, s, r) = (A, f, g)\|_{-1}$$

is a delooping of the dihedral group D_n.

Alternative proof. We would like to provide another proof Proposition 14, which was suggested by an anonymous reviewer. It is based on the idea that in order to show $\Omega\,\mathrm{P}_X \simeq \Omega\,\mathrm{P}_G$, it is enough to show that $U : \mathrm{Set}_G \to \mathrm{Set}_X$ is an *embedding*, i.e. that for every $\alpha, \beta : \mathrm{Set}_G$ the induced function $U^= : (\alpha = \beta) \to (U\alpha = U\beta)$ between path spaces is an equivalence [39, Definition 4.6.1]. It relies on the following results.

Lemma 18. *Given a type $A : \mathcal{U}$, type families $P, Q : A \to \mathcal{U}$ and $f : (a : A) \to P\,a \to Q\,a$, the map $\Sigma A.f : \Sigma A.P \to \Sigma A.Q$ is an embedding if and only if $f\,a : P\,a \to Q\,a$ is an embedding for every $a : A$.*

Proof. By definition, the map $\Sigma A.f$ is an embedding iff for every (a, x) and (a', x') in $\Sigma A.X$, the induced map

$$(a, x) = (a', x') \to (a, f\,a\,x) = (a', f\,a'\,x')$$

is an equivalence. By the characterization of equalities in Σ-types [39, Theorem 2.7.2], this map corresponds to a map

$$(\Sigma(p : a = a').P_p^{\to}(x) = x') \to (\Sigma(p : a = a').Q_p^{\to}(f\,a\,x) = f\,a'\,x')$$

By [39, Theorem 4.7.7], this is an equivalence if and only if the fiber map

$$(P_p^{\to}(x) = x') \to (Q_p^{\to}(f\,a\,x) = f\,a'\,x')$$

is an equivalence for every $p : a = a'$. By path induction, this is true if and only if

$$(f\,a)^= : x = x' \to f\,a\,x = f\,a\,x'$$

is an equivalence for all $a : A$, and $x, x' : Xa$. By definition, this is the requirement that $f\,a$ is an embedding for all $a : A$. ☐

Lemma 19. *Given a morphism of groups $f : H \to_{\mathrm{Grp}} K$, we write $\mathrm{Set}_f : \mathrm{Set}_K \to \mathrm{Set}_H$ for the function induced by precomposition. If f is surjective then Set_f is an embedding.*

Proof. Consider the map

$$F : (A : \mathrm{Set}) \to (K \to_{\mathrm{Grp}} \operatorname{Aut} A) \to (H \to_{\mathrm{Grp}} \operatorname{Aut} A)$$

obtained by precomposition by f. Since f is surjective we have that $F\,A$ is an embedding for every set A [39, Lemma 10.1.4]. By Lemma 18, we deduce that $\Sigma\,\mathrm{id}_{\mathrm{Set}}.F$, which is Set_f, is an embedding. ☐

Lemma 20. *The map $\mathrm{Set}_\iota : \mathrm{Set}_{X^*} \to \mathrm{Set}_X$ is an embedding.*

Proof. By universal property of X^*, given a set A, the map

$$(X^* \to_{\mathrm{Grp}} \operatorname{Aut} A) \to (X \to \operatorname{Aut} A)$$

obtained by precomposition by ι is an equivalence, and thus an embedding. Moreover, the property of being an isomorphism is a proposition, hence the forgetful map

$$(X \to \operatorname{Aut} A) \to (X \to \operatorname{End} A)$$

is an embedding because its fibers are propositions. Since embeddings are stable under composition, we deduce that the induced map

$$F : (A : \mathrm{Set}) \to (X^* \to_{\mathrm{Grp}} \operatorname{Aut} A) \to (X \to \operatorname{End} A)$$

is such that $F\,A$ is an embedding for every set A. By Lemma 18, the map $\Sigma\,\mathrm{id}_{\mathrm{Set}}\,.F$, which is Set_ι, is thus an embedding. □

Proposition 21. *The map $U : \mathrm{Set}_G \to \mathrm{Set}_X$ is an embedding.*

Proof. Given a morphism of groups $f : H \to K$, we write $\mathrm{Set}_f : \mathrm{Set}_K \to \mathrm{Set}_H$ for the function induced by precomposition. In particular, by definition, we have $U \equiv \mathrm{Set}_\gamma$. The function $\gamma : X \to G$ can be decomposed as $\gamma^* \circ \iota$, and therefore Set_γ can be decomposed as $\mathrm{Set}_\iota \circ \mathrm{Set}_{\gamma^*}$:

$$\begin{tikzcd}
& X^* \arrow[dl, "\gamma^*"'] & \\
G & & X \arrow[ll, "\gamma"] \arrow[ul, "\iota"']
\end{tikzcd}
\qquad
\begin{tikzcd}
& \mathrm{Set}_{X^*} \arrow[dr, "\mathrm{Set}_\iota"] & \\
\mathrm{Set}_G \arrow[ur, "\mathrm{Set}_{\gamma^*}"] \arrow[rr, "U"'] & & \mathrm{Set}_X
\end{tikzcd}$$

By previous lemmas, both maps are embeddings so that U is an embedding by composition. □

5. Delooping using higher inductive types

We now recall an alternative construction in order to construct deloopings of groups, when higher inductive types are available in the theory. We then explain how to refine it when a presentation is known for the group.

Delooping as a higher inductive type. Given a group G, its delooping should have a point $\star$ and a loop for every element of the group. Moreover, we should ensure that the multiplication of G coincides with the concatenation operation on the loop space, and that the type we obtain is a (pointed connected) groupoid. This suggests considering a higher inductive type, noted $\mathrm{K}(G,1)$, with the following constructors

$$\begin{array}{ll}
\star & : \mathrm{K}(G,1) \\
\mathsf{loop} & : G \to \star = \star \\
\mathsf{loop\text{-}comp} & : (x, y : G) \to \mathsf{loop}\,x \cdot \mathsf{loop}\,y = \mathsf{loop}(xy) \\
\mathsf{trunc} & : \mathrm{isGroupoid}(\mathrm{K}(G,1))
\end{array}$$

This construction was first proposed by Finster and Licata. They also showed, using the encode-decode method, that it is a delooping of the original group, i.e. $\Omega\,\mathrm{K}(G,1) = G$, see [25, Theorem 3.2]. Note that we only ask here that loop preserves multiplication (with $\mathsf{loop\text{-}comp}$), because it can be shown that this implies preservation of unit and inverses. In particular, preservation of unit renders superfluous one of the constructors present in the original definition [25]:

Lemma 22 (`EM.loop-id`). *In* $\mathrm{K}(G,1)$*, we have* $\mathsf{loop}\,1 = \mathrm{refl}$.

Proof. Omitting associativity and unitality of path composition, we have

$$\mathsf{loop}\,1 = (\mathsf{loop}\,1)^{-1} \cdot \mathsf{loop}\,1 \cdot \mathsf{loop}\,1 = (\mathsf{loop}\,1)^{-1} \cdot \mathsf{loop}\,1 = \mathrm{refl}$$

where the equality on the middle is derived from $\mathsf{loop\text{-}comp}\,1$. ☐

In the following, we will define a variant of this higher inductive type when the group G is presented, which is smaller and gives rise to computations closer to traditional group theory.

Presentations of groups. Any free group X^* on a set X admits a delooping as a wedge of an X-indexed family of circles. The corresponding type $\bigvee_X \mathrm{S}^1$ can be described as the coequalizer

$$\begin{tikzcd} X \arrow[r, shift left] \arrow[r, shift right] & 1 \arrow[r, dotted] & \bigvee_X \mathrm{S}^1 \end{tikzcd} \tag{5.1}$$

or, equivalently, as the higher inductive type

$$\begin{aligned} \star \quad &: \textstyle\bigvee_X \mathrm{S}^1 \\ \mathsf{loop} &: X \to \star = \star \end{aligned}$$

generated by the two constructors $\star : \bigvee_X \mathrm{S}^1$ and $\mathsf{loop} : X \to \star = \star$.

Proposition 23. *We have* $\Omega \bigvee_X \mathrm{S}^1 = X^*$*, i.e. the above type is a* $\mathrm{B}\,X^*$.

Proof. The fact that $\bigvee_X \mathrm{S}^1$ is a delooping of X^* is not too difficult to show when X has decidable equality, see [39, Exercise 8.2] and [22], but the general case is more involved and was recently proved in [42]: the main issue is to show that this type is a groupoid. ☐

A group *presentation* $\langle X \mid R\rangle$ consists of a set X of *generators*, a set R of *relations*, and two functions $\pi, \pi' : R \to X^*$ respectively associating to a relation its *source* and *target*. We often write $r : u \Rightarrow v$ for a relation r with u as source and v as target. Given such a presentation P, the corresponding *presented* group $[P]$ is the set quotient X^*/R of the free group on X under the smallest congruence identifying the source and the target of every relation $r : R$. This type can be described as the type $[P] \equiv \|X^* /\!\!/ R\|_0$ obtained by taking the set truncation of the coequalizer

$$\begin{tikzcd} R \arrow[r, shift left, "\pi"] \arrow[r, shift right, "\pi'"'] & X^* \arrow[r, dotted, "\kappa"] & X^* /\!\!/ R \end{tikzcd}$$

From this also follows a description of $[P]$ as a higher inductive type:

$$\begin{aligned} \mathsf{word} &: X^* \to [P] \\ \mathsf{rel} &: (r : R) \to \mathsf{word}(\pi(r)) = \mathsf{word}(\pi'(r)) \\ \mathsf{trunc} &: \mathrm{isSet}([P]) \end{aligned}$$

A smaller delooping. Suppose given a group G along with a presentation $P \equiv \langle X \mid R \rangle$, i.e. such that $G = [P]$. We define the type $\mathrm{B}\,P$ as the following higher inductive type:

$$\begin{array}{ll} \star & : \mathrm{B}\,P \\ \mathsf{gen} & : X \to (\star = \star) \\ \mathsf{rel} & : (r : R) \to (\mathsf{gen}^*(\pi(r)) = \mathsf{gen}^*(\pi'(r))) \\ \mathsf{trunc} & : \mathrm{isGroupoid}(\mathrm{B}\,P) \end{array}$$

This type is generated by a point $\star$, then the constructor gen adds a loop $\underline{a} : \star = \star$ for every generator a, the constructor rel adds an equality $\underline{a_1} \cdot \underline{a_2} \cdot \ldots \cdot \underline{a_n} = \underline{b_1} \cdot \underline{b_2} \cdot \ldots \cdot \underline{b_m}$ for each relation $a_1 \ldots a_n \Rightarrow b_1 \ldots b_m$, and the constructor trunc formally takes the groupoid truncation of the resulting type. Note that, because of the presence of $\mathsf{gen}^* : X^* \to (\star = \star)$ in the type of rel, the above inductive type is not accepted as is in standard proof assistants such as Agda. However, a definition can be done in two stages, by first considering $\bigvee_X \mathrm{S}^1$ (i.e. the type generated only by $\star$ and gen), and then defining a second inductive type further quotienting this type (i.e. adding the constructors rel and trunc), see `EM.Delooping`. Also, the definition of gen^* requires the group structure on $\bigvee_X \mathrm{S}^1$: the group operations are easily defined from operations on paths (reflexivity, concatenation, symmetry), but the fact that it is a groupoid is non-trivial (see Proposition 23). Our main result in this section is the following:

Theorem 24 (`EM.theorem`). *Given a presentation $P \equiv \langle X \mid R \rangle$, the type $\mathrm{B}\,P$ is a delooping of the group $[P]$.*

Proof. By induction on $\mathrm{B}\,P$, we can define a function $f : \mathrm{B}\,P \to \mathrm{K}([P], 1)$ such that $f\star \equiv \star$, and $f(\mathsf{gen}\,a) \equiv \mathsf{loop}[a]$ for all $a : X$. It can be shown that f is then such that $f^=(\mathsf{gen}^*\,u) = \mathsf{loop}[u]$, for any $u : X^*$. We can therefore define the image $f^=(r)$ on a relation $r : u \Rightarrow v$ as the composite of equalities

$$f^=(\mathsf{gen}^*\,u) = \mathsf{loop}[u] = \mathsf{loop}[v] = f^=(\mathsf{gen}^*\,v)$$

where the equality in the middle follows from the fact that we have $[u] = [v]$ because of the relation r.

In the other direction, the group morphism $\mathsf{gen}^* : X^* \to \Omega\,\mathrm{B}\,P$ preserves relations (by rel), and thus induces a quotient morphism $g' : [P] \to \Omega\,\mathrm{B}\,P$. We can thus consider the function $g : \mathrm{K}([P], 1) \to \mathrm{B}\,P$ such that $g(\star) = \star$, for $x : [P]$ we have $g^=(\mathsf{loop}\,x) = g'(x)$, and for $x, y : [P]$ the image of $\mathsf{loop\text{-}comp}\,x\,y$ is canonically induced by the fact that g' preserves group multiplication.

Since $\mathrm{K}([P], 1)$ is a groupoid, in order to show that $f(g(x)) = x$ for every $x : \mathrm{K}([P], 1)$, it is enough to show that it holds for $x \equiv \star$, which is the case by definition of f and g, and that this property is preserved under $\mathsf{loop}\,x$ for $x : [P]$, which follows from the fact that we have $f^=(g'(x)) = \mathsf{loop}\,x$ for any $x : [P]$ (this is easily shown by induction on x). Conversely, we have to show that $g(f(x)) = x$ holds for $x : \mathrm{B}\,P$. Again, this is shown by induction on x. □

As an interesting remark, the careful reader will note that the fact that the types X and R are sets does not play a role in the proof: in fact, those assumptions can be dropped here. Also, note that we do not need the choice of a representative in X^* for every element of $[P]$ in order to define the function g from gen^* in the above proof: intuitively, this is because the induced function g does not depend on such a choice of representatives. Finally, we should mention here that a similar result is mentioned as an exercise in [39, Example 8.7.17];

the proof suggested there is more involved since it is based on a generalized van Kampen theorem.

Example 25. The dihedral group D_5, see Example 17, admits the presentation

$$\langle r, s \mid r^4 = srs, sr^2s = r^3, rsr = s, r^3s = sr^2, sr^3 = r^2s, s^2 = 1\rangle$$

Hence, by Theorem 24 we can construct a delooping of D_5 as an higher inductive type generated by two loops (corresponding to r and s) and six 2-dimensional cells (corresponding to the relations). Note that this is much smaller than $\mathrm{K}(D_5, 1)$ (it has 2 instead of 10 generating loops, and 6 instead of 100 relations), thus resulting in shorter proofs when reasoning by induction.

Example 26. Any group G admits a presentation, the *standard presentation*, with one generator $\underline{a}$ for every element $a : G$, and relations $\underline{a}\ \underline{b} = \underline{ab}$ for every pair of generators, as well as $\underline{1} = 1$. By applying Theorem 24, we actually recover the inductive type $\mathrm{K}(G, 1)$ as delooping of G.

6. 2-POLYGRAPHS

In Section 5, we have constructed the higher inductive type $\mathrm{B}\,P$ which is smaller than the standard construction of Finster and Licata [25] for $\mathrm{B}\,G$ when a presentation P for G is known. In this section, we introduce the notion of *2-polygraph* which internalizes the data necessary to describe such higher inductive types and manipulate those. This notion, which originates in the study of strict higher categories [10, 2] has been recently ported to homotopy type theory by Kraus and von Raummer [23] and further studied in [30]. Here, we introduce the notions of generated and presented type of a 2-polygraph, so that the type presented by a 2-polygraph P corresponding to a presentation of a group G is precisely $\mathrm{B}\,P$. We also illustrate the usefulness of 2-polygraphs by introducing Tietze transformations, which allow performing computations on those while preserving the presented type (Theorem 29). These operations can thus be used to transform a presentation into one more amenable to computations (such as a convergent one, see Remark 31).

Polygraphs. A *0-polygraph* P is a set P_0. A *0-sphere* in such a polygraph is a pair of elements of P_0, and we write $\mathrm{S}^0_P \equiv P_0 \times P_0$ for the type of spheres in P.

A *1-polygraph* P consists of a 0-polygraph P_0 together with a function $P_1 : \mathrm{S}^0_P \to \mathrm{Set}$ which to a sphere (x, y) associates the set $P_1(x, y)$ of *1-generators* from x to y (x and y are sometimes respectively referred to as the *source* and *target* of the 1-generator). We write $P^*_1 : \mathrm{S}^0_P \to \mathrm{Set}$ for the function which to a 0-sphere (x, y) associates the set $P^*_1(x, y)$ of *1-cells* from x to y. Those consist of formal composable zig-zags of 1-generators, i.e. formal composites of generators or inverses of those, which can be formally defined as sequences

$$a_1^{\epsilon_1} \cdot a_2^{\epsilon_2} \cdots a_n^{\epsilon_n}$$

where $a_i^{\epsilon_i}$ is either of the form a_i^+ with $a_i : P_1(x_{i-1}, x_i)$ or a_i^- with $a_i : P_1(x_i, x_{i-1})$ for some sequence of 0-generators $x_0, x_1, \ldots, x_n : P_0$, with $x_0 = x$ as *source* and $x_n = y$ as *target*. Given a 0-cell x, we write $\mathrm{id}_x : P^*_1(x, x)$ for the nullary composition (the trivial 1-cell). Given two 1-cells $u : P^*_1(x, y)$ and $v : P^*_1(y, z)$, we write $u *_0 v : P_1(x, z)$ for the 1-cell obtained by concatenation of the two lists. Note that our definition of 1-cells is not "coherent" in the sense that we would expect relations such as $a^- \cdot a^+ = \mathrm{id}$ to hold for any 1-generator a, as well as associated higher coherences. However, we will see that those can be introduced when needed, as explicit 2-cells in the definition of P_2^* below. A *1-sphere* in P is a pair of two 1-cells with the same source and target, and we write

$$\mathrm{S}_P^1 \quad\equiv\quad \Sigma((x,y):\mathrm{S}_P^0).(P_1(x,y)\times P_1(x,y))$$

for the type of 1-sphere in P. We often simply write (u,v) instead of $((x,y),(u,v))$ for a 1-sphere.

A *2-polygraph* P consists of a 1-polygraph $P'=(P_0,P_1)$ as above, together with a function $P_2:\mathrm{S}_P^1\to\mathrm{Set}$ which to a 1-sphere (u,v) associates the set $P_2(u,v)$ of 2-generators of the 2-polygraph.

Remark 27. The definition of polygraphs is "indexed" in the sense that we provide sets of $(n{+}1)$-generators which are indexed by n-spheres. By the Grothendieck duality (see Section 1), we can also provide an equivalent fibered definition, where we provide the (total) set of all $(n{+}1)$-generators along with a function associating their boundary n-sphere.

More explicitly, a *fibered 1-polygraph* consists of a 0-polygraph P_0 together with a set P_1 and a function $\partial:P_1\to\mathrm{S}_P^0$ which to every 1-cell associates its boundary 0-sphere. Similarly, a *fibered 2-polygraph* consists in a fibered 1-polygraph together with a set P_2 and a function $\partial:P_2\to\mathrm{S}_P^1$. In this context, we often write $a:x\to y$ in order to indicate that a is an element of P_1 with $\partial(a)=(x,y)$, and similarly we write $\alpha:u\Rightarrow v$ for $\alpha:P_2$ with $\partial\alpha=(u,v)$.

For instance, any (non-fibered) 1-polygraph P induces a fibered 1-polygraph Q with

$$Q_0\equiv P_0 \qquad\qquad Q_1\equiv\Sigma((x,y):\mathrm{S}_P^0).P_1(x,y)$$

and $\partial:Q_1\to\mathrm{S}_P^0$ is the first projection. Conversely, any fibered 1-polygraph Q induces a non-fibered one P by setting $P_0\equiv Q_0$ and $P_1(x,y)=\mathrm{fib}_\partial(x,y)$. These two operations are mutually inverse, and a similar equivalence can be constructed for 2-polygraphs.

Generated type. Given a 1-polygraph P, the type *generated* by P is the type $\overline{P}$ described as the following higher inductive type

$$\begin{aligned}&\mathsf{pt}\ \ :P_0\to\overline{P}\\&\mathsf{gen}:((x,y):\mathrm{S}_P^0)\to(a:P_1(x,y))\to(\mathsf{pt}\,x=\mathsf{pt}\,y)\end{aligned}$$

We write

$$\mathsf{gen}^*:((x,y):\mathrm{S}_P^0)\to(u:P_1^*(x,y))\to(\mathsf{pt}\,x=\mathsf{pt}\,y)$$

for the expected extension of gen to cells, defined by

$$\mathsf{gen}^*(u\cdot v)=\mathsf{gen}^*(u)\cdot\mathsf{gen}^*(v) \qquad\qquad \mathsf{gen}^*(a^+)=\mathsf{gen}\,a$$

$$\mathsf{gen}^*(\mathrm{id})=\mathrm{refl} \qquad\qquad \mathsf{gen}^*(a^-)=(\mathsf{gen}\,a)^-$$

More abstractly, $\overline{P}$ can be computed as the following coequalizer

$$\begin{tikzcd}\Sigma\,\mathrm{S}_P^0\,.P_1 \arrow[r, shift left, "\pi\pi"] \arrow[r, shift right, "\pi'\pi"'] & P_0 \arrow[r, dotted] & \overline{P}\end{tikzcd}$$

where the two maps $\Sigma\,\mathrm{S}_P^0\,.P_1\to P_0$ respectively send a pair $((x,y),a)$ to x and y: given a 1-cell, we identify its source and target in $\overline{P}$.

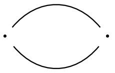
Given a 2-polygraph P, the type $\overline{P}$ *generated* by P is the type generated by the higher inductive type

$$\begin{aligned}&\mathsf{pred}:\overline{P'}\to\overline{P}\\&\mathsf{rel}\ \ \ :((u,v):\mathrm{S}_P^1)\to(\alpha:P_2(u,v))\to(\mathsf{gen}^*(u)=\mathsf{gen}^*(v))\end{aligned}$$

Note that, by the first constructor, we have a canonical inclusion in $\overline{P}$ of the type generated (in the above sense) by the underlying 1-polygraph P'.

In order to provide an alternative definition of the type generated by a polygraph as a colimit, we first need to introduce some notations. We write I for the *interval* type, generated by two points and a path between them. This type is of course contractible, so that we could use the equivalent type 1 instead, but we believe that its use clarifies notations and intuitions. We also write O for the *globe* type, which is defined as the pushout on the left and corresponds to the space pictured on the right

$$\begin{tikzcd} 2 \arrow[r, "i"] \arrow[d, "i"'] & I \arrow[d] \\ I \arrow[r] & O \arrow[ul, phantom, "\ulcorner", very near start] \end{tikzcd}$$

$O \equiv$  (6.1)

Above, the map $i : 2 \to I$ is the canonical inclusion (sending the elements of 2 to the endpoints of I). One should think of O as a two intervals glued one their endpoints, thus forming a circle: these intervals are sometimes respectively referred to as the northern and southern hemispheres. It is easily shown that O equivalent to S^1. By the universal property of the pushout, any two maps $f, g : I \to X$ together with an identification $f \circ i = g \circ i$ induces a map denoted $f + g : O \to X$.

With those notations, the type $\overline{P}$ generated by a 2-polygraph P can be described as the pushout

$$\begin{tikzcd} O \times \Sigma\,\mathrm{S}^1_P\,.P_2 \arrow[r, "\pi'"] \arrow[d] & \Sigma\,\mathrm{S}^1_P\,.P_2 \arrow[d] \\ \overline{P'} \arrow[r] & \overline{P} \arrow[ul, phantom, "\ulcorner", very near start] \end{tikzcd} \tag{6.2}$$

where P' is the underlying 1-polygraph of P, the topmost horizontal map is the second projection and the leftmost vertical map is obtained by curryfication from the map

$$\Sigma\,\mathrm{S}^1_P\,.P_2 \to (O \to \overline{P'})$$
$$(((x,y),(u,v)),\alpha) \to \mathsf{gen}^*\,u + \mathsf{gen}^*\,v$$

sending a 2-generator to the globe in $\overline{P'}$ corresponding to its boundary. This construction of $\overline{P}$ can be summarized in the following diagram:

$$\begin{tikzcd} & & O \times \Sigma\,\mathrm{S}^1_P\,.P_2 \arrow[r, "\pi'"] \arrow[d] & \Sigma\,\mathrm{S}^1_P\,.P_2 \arrow[d] \\ \Sigma\,\mathrm{S}^0_P\,.P_1 \arrow[r, shift left, "\pi\pi"] \arrow[r, shift right, "\pi'\pi"'] & P_0 \arrow[r] & \overline{P'} \arrow[r] & \overline{P} \arrow[ul, phantom, "\ulcorner", very near start] \end{tikzcd} \tag{6.3}$$

Note that this is an instance of a cellular type in the sense of [8].

Presented type. The type *presented* by a 2-polygraph P is $\|\overline{P}\|_1$.

When $P_0 \equiv 1$ contains only one element, a (fibered) 2-polygraph P encodes precisely the data of a presentation $\langle X \mid R \rangle$ of a group G with $X \equiv P_1$ as generators and $R \equiv P_2$ as relations, each element $\alpha : P_2$ with $\partial(\alpha) = (u, v)$ encoding a relation identifying u and v. The type presented by this polygraph is then precisely the one we described in Section 5.

Conversely, the notion of 2-polygraph can be thought of as a mild generalization of the notion of group presentation, where we allow multiple 0-cells.

Proposition 28. *Given a group G with a presentation $\langle X \mid R\rangle$, consider the fibered 2-polygraph P with $P_0 \equiv 1$, $P_1 \equiv X$, $P_2 \equiv R$ and $\partial(u,v) = (u,v)$. The type presented by P is $\mathrm{B}\,P = \|\overline{P}\|_1$.*

By analogy to P_1^*, we can define a function $P_2^* : \mathrm{S}^1_P \to \mathrm{Set}$, such that the elements of $P_2^*(u,v)$ are called the *2-cells* (or *relations*) from u to v in P. It is defined as the smallest family of sets $P_2^* : \mathrm{S}^1_P \to \mathrm{Set}$ such that

- for any 1-sphere (u,v), every 2-generator $\alpha : P_2(u,v)$ induces an element in $P_2^*(u,v)$, still noted α,

$$\begin{tikzcd} x \arrow[r, bend left=50, "u", ""{name=U, below}] \arrow[r, bend right=50, "v"', ""{name=D}] & y \arrow[Rightarrow, from=U, to=D, "\alpha"'] \end{tikzcd}$$

- for any 1-spheres $((x,y),(u,u'))$ and $((y,z),(v,v'))$, any 2-cells $\alpha : P_2^*(u,u')$ and $\beta : P_2^*(v,v')$ induce a 2-cell $\alpha *_0 \beta : P_2^*(u *_0 v, u' *_0 v')$,

$$\begin{tikzcd} x \arrow[r, bend left=50, "u", ""{name=U1, below}] \arrow[r, bend right=50, "u'"', ""{name=D1}] & y \arrow[r, bend left=50, "v", ""{name=U2, below}] \arrow[r, bend right=50, "v'"', ""{name=D2}] & z \arrow[Rightarrow, from=U1, to=D1, "\alpha"'] \arrow[Rightarrow, from=U2, to=D2, "\beta"'] & \rightsquigarrow & x \arrow[rr, bend left=40, "u*_0v", ""{name=U3, below}] \arrow[rr, bend right=40, "u'*_0v'"', ""{name=D3}] & & z \arrow[Rightarrow, from=U3, to=D3, "\alpha*_0\beta"'] \end{tikzcd}$$

- for any 1-spheres (u,v) and (v,w), any 2-cells $\alpha : P_2^*(u,v)$ and $\beta : P_2^*(v,w)$ induce a 2-cell $\alpha *_1 \beta : P_1^*(u,w)$,

$$\begin{tikzcd} x \arrow[r, bend left=60, "u", ""{name=U, below}] \arrow[r, "v" description, ""{name=M}] \arrow[r, bend right=60, "w"', ""{name=D}] & y \arrow[Rightarrow, from=U, to=M, "\alpha"'] \arrow[Rightarrow, from=M, to=D, "\beta"'] & \rightsquigarrow & x \arrow[r, bend left=50, "u", ""{name=U2, below}] \arrow[r, bend right=50, "w"', ""{name=D2}] & y \arrow[Rightarrow, from=U2, to=D2, "\alpha*_1\beta"'] \end{tikzcd}$$

- any 1-cell $u : P_1^*(x,y)$ induces a 1-cell $\mathrm{id}_u : P_2^*(u,u)$,

$$\begin{tikzcd} x \arrow[r, bend left=50, "u", ""{name=U, below}] \arrow[r, bend right=50, "u"', ""{name=D}] & y \arrow[Rightarrow, from=U, to=D, "\mathrm{id}_u"'] \end{tikzcd}$$

- for any 1-sphere (u,v), any 2-cell $\alpha : P_2^*(u,v)$ induces a 2-cell $\alpha^{-1} : P_2^*(v,u)$,

$$\begin{tikzcd} x \arrow[r, bend left=50, "u", ""{name=U, below}] \arrow[r, bend right=50, "v"', ""{name=D}] & y \arrow[Rightarrow, from=U, to=D, "\alpha"'] & \rightsquigarrow & x \arrow[r, bend left=50, "u", ""{name=U2, below}] \arrow[r, bend right=50, "v"', ""{name=D2}] & y \arrow[Rightarrow, from=D2, to=U2, "\alpha^{-1}"] \end{tikzcd}$$

- for any 1-generator $a : P_1(x,y)$ induces a pair of 2-cells $\lambda_a : P_2^*(a^- \cdot a^+, \mathrm{id}_y)$ and $\rho_a : P_2^*(a^+ \cdot a^-, \mathrm{id}_x)$:

$$\begin{tikzcd} & x \arrow[dr, bend left, "a^+"] \arrow[d, Rightarrow, "\lambda_a"] & & & & y \arrow[dr, bend left, "a^-"] \arrow[d, Rightarrow, "\rho_a"] & \\ y \arrow[ur, bend left, "a^-"] \arrow[rr, bend right=20, "\mathrm{id}_y"'] & {} & y & & x \arrow[ur, bend left, "a^+"] \arrow[rr, bend right=20, "\mathrm{id}_x"'] & {} & x \end{tikzcd}$$

As for the definition of P_1^*, we would expect some laws to hold (associativity of composition, exchange law, etc.), but those are not needed here because we only consider the mere existence of a 2-cell with given 1-cell as boundary. They would however be required if we were to define 3-polygraphs.

Tietze transformations. We now define operations on polygraphs, called *Tietze transformations*, turning a 2-polygraph P into another 2-polygraph Q. Those are of the following three kinds.

(T0) Starting from P together with $a : P_0$, we construct the polygraph Q with

$$Q_0 = P_0 \sqcup \{\star\} \qquad Q_1(x,y) = \begin{cases} P_1(x,y) & \text{if } x : P_0 \text{ and } y : P_0 \\ 1 & \text{if } x \equiv a \text{ and } y \equiv \star \\ 0 & \text{otherwise} \end{cases}$$

$$Q_2(u,v) = \begin{cases} P_2(u,v) & \text{if } u, v : P_1^*(x,y) \text{ for some } (x,y) : \mathrm{S}^0_P \\ 0 & \text{otherwise} \end{cases}$$

(T1) Starting from P together with $w : P_1^*(a,b)$ for some $(a,b) : \mathrm{S}^0_P$, we construct the polygraph Q with

$$Q_0 = P_0 \qquad Q_1(x,y) = \begin{cases} P_1(x,y) \sqcup \{\star\} & \text{if } x \equiv a \text{ and } y \equiv b \\ P_1(x,y) & \text{otherwise} \end{cases}$$

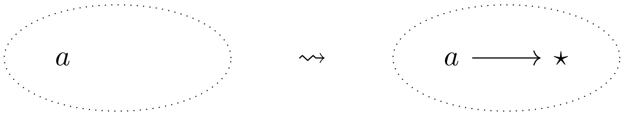
$$Q_2(u,v) = \begin{cases} P_2(u,v) & \text{if } u, v : P_1^*(x,y) \text{ for some } (x,y) : \mathrm{S}^0_P \\ 1 & \text{if } u \equiv w \text{ and } v \equiv \star \\ 0 & \text{otherwise} \end{cases}$$

(T2) Starting from P together with $\alpha : P_2^*(w,w')$ for some $(w,w') : \mathrm{S}^1_P$, we construct the polygraph Q with

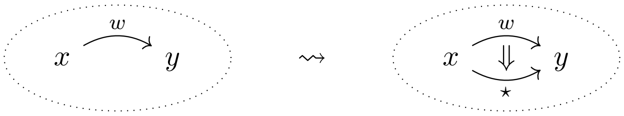
$$Q_0 = P_0 \qquad Q_1 = P_1 \qquad Q_2(u,v) = \begin{cases} P_2(u,v) \sqcup \{\star\} & \text{if } u \equiv w \text{ and } v \equiv w' \\ P_2(u,v) & \text{otherwise} \end{cases}$$

Graphically, we have that

(T0) a transformation (T0) consists, given a 0-generator $a : P_0$, in adding a new 0-generator $\star$ and a 1-generator from a to $\star$:



(T1) a transformation (T1) consists, given a 1-cell $w : P_1^*(x,y)$, in adding a new 1-generator $\star$ together with a 2-generator from w to $\star$:



(T2) a transformation (T2) consists, given a 2-cell $\alpha : P_2^*(w, w')$, in adding a new 2-generator from w to w':

$$\begin{tikzcd} x \arrow[r, bend left, "w"] \arrow[r, bend right, "w'"'] \arrow[r, phantom, "\alpha\Downarrow"] & y \end{tikzcd} \quad\rightsquigarrow\quad \begin{tikzcd} x \arrow[r, bend left, "w"] \arrow[r, bend right, "w'"'] \arrow[r, phantom, "\alpha\Downarrow\Downarrow\star"] & y \end{tikzcd}$$

Tietze equivalence. Two 2-polygraphs are *Tietze equivalent* when they are related by the smallest equivalence relation identifying two 2-polygraphs related by a Tietze transformation.

Theorem 29. *Two Tietze equivalent 2-polygraphs P and Q present the same types.*

Proof. It is enough to show that for each Tietze transformation from P to Q, the polygraphs present the same type, i.e. we have $\|\overline{P}\|_1 = \|\overline{Q}\|_1$. For transformations (T0) and (T1), we will actually be able to show the stronger property that we have $\overline{P} = \overline{Q}$.

(T0) By (6.3), we have that $\overline{Q}$ is the colimit of

$$\begin{tikzcd} & & O \times \Sigma\,\mathrm{S}^1_P\,.P_2 \arrow[r] \arrow[d] & \Sigma\,\mathrm{S}^1_P\,.P_2 \arrow[d] \\ \Sigma\,\mathrm{S}^0_P\,.P_1 \sqcup 1 \arrow[r, shift left] \arrow[r, shift right] & P_0 \sqcup 1 \arrow[r] & \overline{Q'} \arrow[r] & \overline{Q} \arrow[ul, phantom, "\ulcorner", very near start] \end{tikzcd} \tag{6.4}$$

By commutation of colimits (see [5, Lemma 1.8.3]), we have that $\overline{Q'}$ is the coequalizer

$$\begin{tikzcd} 1 \arrow[r, shift left, "a"] \arrow[r, shift right, "\star"'] & \overline{P'} \sqcup 1 \arrow[r, dotted] & \overline{Q'} \end{tikzcd}$$

or equivalently the pushout

$$\begin{tikzcd} 1 \arrow[r, "a"] \arrow[d, "\mathrm{id}_1"'] & \overline{P'} \arrow[d] \\ 1 \arrow[r] & \overline{Q'} \arrow[ul, phantom, "\ulcorner", very near start] \end{tikzcd}$$

Moreover, the map $O \times \Sigma\,\mathrm{S}^1_P\,.P_2 \to \overline{Q'}$ factors through $\overline{P'}$ (because the boundary of a 2-generator in P_2 lies in P_1^*, as opposed to Q_1^*), so that the definition (6.4) reduces to the following sequence of pushouts:

$$\begin{tikzcd} & O \times \Sigma\,\mathrm{S}^1_P\,.P_2 \arrow[r] \arrow[d] & \Sigma\,\mathrm{S}^1_P\,.P_2 \arrow[d] \\ 1 \arrow[r, "a"] \arrow[d, "\mathrm{id}_1"'] & \overline{P'} \arrow[r] \arrow[d] & \overline{P} \arrow[d] \arrow[ul, phantom, "\ulcorner", very near start] \\ 1 \arrow[r] & \overline{Q'} \arrow[r] \arrow[ul, phantom, "\ulcorner", very near start] & \overline{Q} \arrow[ul, phantom, "\ulcorner", very near start] \end{tikzcd}$$

We thus have the pushout

$$\begin{tikzcd} 1 \arrow[r, "a"] \arrow[d, "\mathrm{id}_1"'] & \overline{P} \arrow[d] \\ 1 \arrow[r] & \overline{Q} \arrow[ul, phantom, "\ulcorner", very near start] \end{tikzcd}$$

from which we deduce that $\overline{P} = \overline{Q}$ and thus $\|\overline{P}\|_1 = \|\overline{Q}\|_1$.

(T1) By (6.3), $\overline{Q}$ is the colimit

$$\begin{tikzcd}
 & & O \times (\Sigma\,\mathrm{S}^1_P\,.P_2 \sqcup 1) \arrow[r] \arrow[d] & \Sigma\,\mathrm{S}^1_P\,.P_2 \arrow[d] \\
\Sigma\,\mathrm{S}^0_P\,.P_1 \sqcup 1 \arrow[r, shift left] \arrow[r, shift right] & P_0 \arrow[r] & \overline{Q'} \arrow[r] & \overline{Q} \arrow[ul, phantom, "\ulcorner", very near start]
\end{tikzcd}$$

By a similar reasoning as above on colimits, we obtain that $\overline{Q}$ is the pushout

$$\begin{tikzcd}
O \arrow[r, "\star+w"] \arrow[d] & \overline{P}/(a=b) \arrow[d] \\
1 \arrow[r] & \overline{Q} \arrow[ul, phantom, "\ulcorner", very near start]
\end{tikzcd} \tag{6.5}$$

where $\overline{P}/(a=b)$ is the pushout

$$\begin{tikzcd}
2 \arrow[r, "a+b"] \arrow[d] & \overline{P} \arrow[d] \\
I \arrow[r] & \overline{P}/(a=b) \arrow[ul, phantom, "\ulcorner", very near start]
\end{tikzcd}$$

We write $\star : a = b$ for the path in $\overline{P}/(a=b)$ added by the colimit. We can construct the following diagram:

$$\begin{tikzcd}
2 \arrow[r] \arrow[d] & I \arrow[r, "w"] \arrow[d] & \overline{P} \arrow[d] \\
I \arrow[r] & O \arrow[r, "\star+w"'] \arrow[ul, phantom, "\ulcorner", very near start] & \overline{P}/(a=b)
\end{tikzcd}$$

The left square is a pushout by construction, see (6.1), and the outer rectangle is a pushout because the top map sends the two points of 2 to a and b respectively, and the bottom one sends the interval to the path $\star$ in $\overline{P}/(a=b)$, see (6.5). Therefore, the right square is also a pushout and, combining this with the definition of $\overline{Q}$, we thus have the following pushout diagram:

$$\begin{tikzcd}
I \arrow[r, "w"] \arrow[d] & \overline{P} \arrow[d] \\
O \arrow[r, "\star+w"] \arrow[d] & \overline{P}/(a=b) \arrow[d] \arrow[ul, phantom, "\ulcorner", very near start] \\
1 \arrow[r] & \overline{Q} \arrow[ul, phantom, "\ulcorner", very near start]
\end{tikzcd}$$

To sum up, replacing I by 1 which is equivalent, $\overline{Q}$ can be obtained from $\overline{P}$ as the pushout

$$\begin{tikzcd}
1 \arrow[r, "w"] \arrow[d, "\mathrm{id}"'] & \overline{P} \arrow[d] \\
1 \arrow[r] & \overline{Q} \arrow[ul, phantom, "\ulcorner", very near start]
\end{tikzcd}$$

from which we deduce that $\overline{P} = \overline{Q}$ and thus $\|\overline{P}\|_1 = \|\overline{Q}\|_1$.

(T2) By (6.3), $\overline{Q}$ is the colimit

$$\begin{tikzcd}
& & O \times (\Sigma\,\mathrm{S}^1_P\,.P_2 \sqcup 1) \arrow[r] \arrow[d] & \Sigma\,\mathrm{S}^1_P\,.P_2 \arrow[d] \\
\Sigma\,\mathrm{S}^0_P\,.P_1 \arrow[r, shift left] \arrow[r, shift right] & P_0 \arrow[r] & \overline{Q'} \arrow[r] & \overline{Q} \arrow[ul, phantom, "\ulcorner", very near start]
\end{tikzcd}$$

By reasoning on colimits, we find that $\overline{Q}$ is the pushout

$$\begin{tikzcd}
O \arrow[r, "w+w'"] \arrow[d] & \overline{P} \arrow[d] \\
1 \arrow[r] & \overline{Q} \arrow[ul, phantom, "\ulcorner", very near start]
\end{tikzcd} \tag{6.6}$$

and we are in the situation

$$\begin{tikzcd}
O \arrow[r] \arrow[d] \arrow[rr, bend left, "w+w'"] & 1 \arrow[r] \arrow[d] & \overline{P} \arrow[d] \\
1 \arrow[r] & \mathrm{S}^2 \arrow[r] \arrow[ul, phantom, "\ulcorner", very near start] & \overline{Q}
\end{tikzcd}$$

where the outer square is a pushout, see (6.6), and the left square is a pushout (this is the definition of S^2 as the suspension of S^1). The square on the right is thus also a pushout. Since 1-truncation is left adjoint it preserves pushouts [39, Theorem 7.4.12] and we have a pushout diagram

$$\begin{tikzcd}
1 \arrow[r] \arrow[d] & \|\overline{P}\|_1 \arrow[d] \\
\|\,\mathrm{S}^2\,\|_1 \arrow[r] & \|\overline{Q}\|_1 \arrow[ul, phantom, "\ulcorner", very near start]
\end{tikzcd}$$

which shows that $\|\overline{P}\|_1 = \|\overline{Q}\|_1$ (since $\|\,\mathrm{S}^2\,\|_1 = 1$, the left map is an identity). □

Example 30. Consider the fibered 2-polygraph P with $P_0 = \{\star\}$, $P_1 = \{a, b\}$, $P_2 = \{\beta\}$ with

$$\partial(\beta) = (aba, bab)$$

i.e.

$$\begin{tikzcd}
& \star \arrow[rr, "b"] & & \star \arrow[dr, bend left, "a"] & \\
\star \arrow[ur, bend left, "a"] \arrow[dr, bend right, "b"'] & & {} & & \star \\
& \star \arrow[rr, "a"'] & & \star \arrow[ur, bend right, "b"'] &
\arrow[Rightarrow, from=1-3, to=3-3, shorten=4mm, "\beta"']
\end{tikzcd}$$

Using a more traditional notation for presentations, this polygraph can be noted as

$$P \quad \equiv \quad \langle \star \mid a, b \mid aba = bab \rangle$$

It is known that $\|\overline{P}\|_1$ is a delooping of B_3, the group of braids with three strands. We have the following series of Tietze transformations:

(T1) we add a 1-generator $c : \star = \star$ and a 2-generator $c = ba$:

$$\langle \star \mid a, b, c \mid aba = bab, c = ba \rangle$$

(T2) we add the derivable relation $ac = cb$:

$$\langle \star \mid a, b, c \mid aba = bab, ac = cb, c = ba\rangle$$

(T2) (backward), we remove the derivable relation $aba = bab$:

$$\langle \star \mid a, b, c \mid ac = cb, c = ba\rangle$$

We finally deduce by Theorem 29 that this last 2-polygraph also presents a delooping of B_3.

Remark 31. The Knuth-Bendix completion algorithm [21] which iteratively adds relation in order to hopefully compute a convergent presentation is also based on Tietze transformations (T2), see [2], which explains why it also preserves the presented type. For instance, it can be used to compute the following convergent presentation of the group B_3 of Example 30:

$$\langle \star \mid a, b, c \mid ccb = acc, bcb = cc, ca = bc, baca = cac, bab = aba, ab = c\rangle$$

However, note that we first need to use a transformation (T1) in order to add the 1-generator c, without it no convergent presentation of B_3 can be found (on the generators a and b only) [20].

Remark 32. In group theory [38] (as well as in polygraph theory [2]), it is known that Tietze transformations are *complete* in the sense that we have a reciprocal to the correctness property (i.e. the analogous of Theorem 29): if two presentations present the same group then they are Tietze equivalent. We expect that a similar property can be shown for 2-polygraphs, but this is left for future works. A proof of completeness in the case of 1-polygraphs can be found in [30].

7. Cayley graphs

Throughout the section, we fix a group G together with a generating set X. We have seen in Section 5 that a delooping of G can be obtained by further homotopy quotienting a delooping of X^*. The kernel of the map $\gamma^* : X^* \to G$ measures the defect of X^* from being G, which corresponds to the relations of the group. We show here that, under the delooping operation, those relations are precisely encoded by the Cayley graph [12, 28], a classical and useful construction in group theory which can be associated to any generated group.

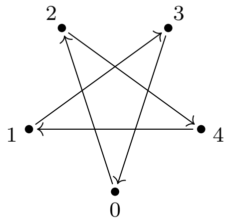
The *Cayley graph* of G, with respect to the generating set X, is the directed graph whose vertices are the elements of G, and such that for every vertex $g : G$ and generator $x : X$, we have an edge $g \to gx$. In homotopy type theory, it is thus natural to represent it as the higher inductive type $\mathrm{C}(X, G)$ defined as

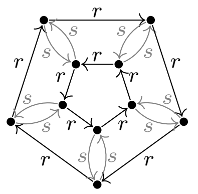
$$\begin{aligned} &\mathsf{vertex} : G \to \mathrm{C}(X, G) \\ &\mathsf{edge} \ \ : (g : G)(x : X) \to \mathsf{vertex}\, g = \mathsf{vertex}(gx) \end{aligned}$$

Example 33. Consider the cyclic group $\mathbb{Z}_5$ (with 2 as generator) and the dihedral group D_5 (with r and s as generators, see also Example 17) which can be presented by

$$\mathbb{Z}_5 = \langle a \mid a^5 = 1\rangle \qquad\qquad D_5 = \langle r, s \mid r^5 = 1, s^2 = 1, rsrs = 1\rangle$$

The associated Cayley graphs are respectively





All the previous constructions can be reformulated in terms of polygraphs. First note that the generating set can be understood as a 1-polygraph P with $P_0 \equiv 1$ and $P_1 \equiv X$, so that $\mathrm{B}\,X^*$ is precisely $\overline{P}$. Similarly, from their very definition, Cayley graphs are generated by polygraphs. Namely, it can be observed that

Proposition 34. *Consider the fibered 1-polygraph Q with*

$$Q_0 \equiv G \qquad\qquad Q_1 \equiv G \times X \qquad\qquad \begin{aligned} \partial : Q_1 &\to \mathrm{S}^0_Q \\ (g, x) &\mapsto (g, gx) \end{aligned}$$

The Cayley graph of G is precisely its generated type:

$$C(X, G) \quad = \quad \overline{Q}$$

Our main result in this section is that Cayley graphs satisfy the following property.

Theorem 35 (`Cayley.Cayley-ker`)**.** *The type $\mathrm{C}(X, G)$ is the kernel of the function*

$$\mathrm{B}\,\gamma^* : \mathrm{B}\,X^* \to \mathrm{B}\,G$$

induced by γ, i.e. we have

$$\mathrm{C}(X, G) \quad = \quad \Sigma(x : \mathrm{B}\,X^*).(\star = \mathrm{B}\,\gamma^*(x)).$$

Proof. We consider the type family

$$\begin{aligned} F : \mathrm{B}\,X^* &\to \mathcal{U} \\ x &\mapsto (\star = \mathrm{B}\,\gamma^*(x)) \end{aligned}$$

Remember that $\mathrm{B}\,X^*$ admits a description as a coequalizer, see (5.1) and Proposition 23:

$$\begin{tikzcd} X \arrow[r, shift left] \arrow[r, shift right] & 1 \arrow[r, dotted] & \mathrm{B}\,X^* \end{tikzcd}$$

Hence, by the flattening lemma for coequalizers (see Section 1 and [39, Section 6.12]), we have a coequalizer of total spaces

$$\begin{tikzcd} \Sigma X.F(\star) \arrow[r, shift left, "(x,p)\mapsto(\star,p)"] \arrow[r, shift right, "(x,p)\mapsto(\star,F^{\rightarrow}_p(x))"'] & \Sigma 1.F(\star) \arrow[r, dotted] & \Sigma\,\mathrm{B}\,X^*.F \end{tikzcd}$$

By using the properties of transport in path spaces [39, Theorem 2.11.3], it can be shown that the bottom map sends (x, p) to $(\star, p \cdot \mathrm{B}\,\gamma^*(x))$. Moreover, $\mathrm{B}\,\gamma^*$ is pointed, so $F(\star)$ is equal to $\Omega\,\mathrm{B}\,G$, i.e. G, and we have the following coequalizer:

$$\begin{tikzcd} X \times G \arrow[r, shift left, "(x,g)\mapsto g"] \arrow[r, shift right, "(x,g)\mapsto gx"'] & G \arrow[r, dotted] & \Sigma\,\mathrm{B}\,X^*.F \end{tikzcd}$$

If follows that $\Sigma\,\mathrm{B}\,X^*.F$ consists in $|G|$ points and a path $g = gx$ for each pair $(x, g) : X \times G$, and is therefore equal to the Cayley graph $\mathrm{C}(X, G)$. □

The above result can be interpreted as stating that we have a fiber sequence

$$\mathrm{C}(X, G) \longrightarrow \mathrm{B}\, X^* \xrightarrow{\mathrm{B}\,\gamma^*} \mathrm{B}\, G$$

Under the Grothendieck duality, the map $\mathrm{B}\,\gamma^* : \mathrm{B}\, X^* \to \mathrm{B}\, G$ corresponds to a type family $\phi : \mathrm{B}\, G \to \mathcal{U}$ with $\phi(\star) = \mathrm{C}(X, G)$, which means that it encodes an action of G on the Cayley graph (which is the canonical one, sending $h \cdot g$ to hg). Moreover, we have $\mathrm{B}\, X^* = \Sigma(\mathrm{B}\, G).\phi$, meaning that $\mathrm{B}\, X^*$ can be understood as the homotopy quotient of the Cayley graph under this action. This point of view is developed in [31, Proposition 16]. Moreover, the map $\mathrm{B}\, X^*$ is an approximation of $\mathrm{B}\, G$ in the following sense:

Proposition 36 (`Cayley.Cayley-connected`). *The Cayley graph* $\mathrm{C}(X, G)$ *is connected and the map* $\mathrm{B}\, X^* \to \mathrm{B}\, G$ *is thus* 0*-connected.*

Proof. We consider $\mathrm{C}(X, G)$ as pointed by the point corresponding to the neutral element of G. It is enough to show that for every point $x : \mathrm{C}(X, G)$ there merely exists a path $\star = x$, which is easily done by induction on x using the fact that X is generating. □

Relations. The long exact sequence of homotopy groups induced by the above fiber sequence [39, Theorem 8.4.6] implies in particular that we have the following short exact sequence of groups

$$1 \longrightarrow \Omega\, \mathrm{C}(X, G) \longrightarrow X^* \longrightarrow G \longrightarrow 1$$

which shows that $\Omega\, \mathrm{C}(X, G)$ is the (free) group encoding relations of G with respect to X. Indeed, we have that $\mathrm{C}(X, G) = \mathrm{B}\, R^*$ where R is a choice of $|G| \times (|X| - 1) + 1$ relations: those are the loops in the Cayley graph after contracting $|G| - 1$ edges to obtain a wedge of circles. In some sense, Theorem 35 provides an internalization of the fact that G is presented by $\langle X \mid R\rangle$, contrasting with the point of view developed in Section 5.

The Cayley complex. We now explain that we can extend the previous construction in higher dimensions in order to define internally a type corresponding to the classical *Cayley complex* [28].

Consider a group G together with a presentation $\langle X \mid R\rangle$, and write P for the 2-polygraph corresponding to this presentation (by Proposition 28). We write $\mathrm{B}_2\, P$ for the 2-skeleton of the type $\mathrm{B}\, P$ defined in Section 5, i.e. the higher inductive type with generators

$$\begin{aligned} \star &: \mathrm{B}\, P \\ \mathsf{gen} &: X \to (\star = \star) \\ \mathsf{rel} &: (r : R) \to (\mathsf{gen}^*(\pi(r)) = \mathsf{gen}^*(\pi'(r))) \end{aligned}$$

Alternatively, $\mathrm{B}_2\, P$ is precisely the type $\overline{P}$ generated by the 2-polygraph P. This type can be considered as an approximation of $\mathrm{B}\, P$: this is “almost” $\mathrm{B}\, P$ excepting we are lacking the groupoid truncation. In particular, this type has the same first two homotopy groups as $\mathrm{B}\, P$. Below, we write $\mathrm{B}_1\, P$ for the type generated by the 1-polygraph of P (note that if we write $X \equiv P_1$, then $\mathrm{B}_1\, P$ is simply another notation for the type $\mathrm{B}\, X^*$ studied in previous section).

Lemma 37. *The type* $\mathrm{B}_2\, P$ *is connected and we have* $\pi_1(\mathrm{B}_2\, P) = G$.

Proof. By definition, the type $\mathrm{B}_2\, P$ is a quotient of the type $\mathrm{B}_1\, P$ (under the relations specified rel, corresponding to P_2). The latter being connected by Proposition 36, the former also is. Moreover, since $\mathrm{B}\,G$ is the groupoid truncation of $\mathrm{B}_2\,P$, they have the same fundamental group. Namely,

$$\pi_1(\mathrm{B}_2\,P) \quad\equiv\quad \|\star \overset{\mathrm{B}_2\,P}{=} \star\|_0 \quad=\quad (|\star|_1 \overset{\|\mathrm{B}_2\,P\|_1}{=} |\star|_1) \quad\equiv\quad \pi_1(\mathrm{B}\,G) \quad=\quad G$$

where the first non-definitional equality is due to [39, Theorem 7.3.12]. □

We would like to "measure the difference" between the two types $\mathrm{B}_2\,P$ and $\mathrm{B}\,G$. Consider the canonical map

$$\phi : \mathrm{B}_2\,P \to \mathrm{B}\,G$$

which is given by sending every path $\mathsf{gen}\,x$ to the loop in $\mathrm{B}\,G$ corresponding to x and each cell $\mathsf{rel}\,r$ to the cell in $\mathrm{B}\,G$ witnessing for the fact that the relation r is satisfied in G. Alternatively, this map is also induced by the fact that we have a definition of $\mathrm{B}\,G$ as a higher inductive type with more generators than $\mathrm{B}_2\,P$ (namely the constructor trunc which corresponds to taking the groupoid truncation), i.e. we have $\mathrm{B}\,G = \|\,\mathrm{B}_2\,G\|_1$ and ϕ is the map

$$|-|_1 : \mathrm{B}_2\,P \to \|\,\mathrm{B}_2\,G\|_1 \tag{7.1}$$

Definition 38. The *Cayley complex* $\mathrm{C}\,P$ associated to the presentation P is the inductive type defined by

$$\begin{array}{ll} \mathsf{vertex} & : G \to \mathrm{C}\,P \\ \mathsf{edge} & : (g : G)(x : X) \to \mathsf{vertex}\,g = \mathsf{vertex}(gx) \\ \mathsf{cell} & : (g : G)(r : R) \to (\mathsf{edge}\,g)^*(\pi(r)) = (\mathsf{edge}\,g)^*(\pi'(r)) \end{array}$$

where, given $g : G$ and $u \equiv x_1x_2\ldots x_n : X^*$, we have that $(\mathsf{edge}\,g)^*\,u$ is the path

$$\mathsf{v}\,g \overset{\mathsf{e}\,g\,x_1}{=\!=} \mathsf{v}(gx_1) \overset{\mathsf{e}(gx_1)\,x_2}{=\!=\!=\!=} \mathsf{v}(gx_1x_2) \overset{\ldots}{=} \cdots \overset{\ldots}{=} \mathsf{v}(gx_1\ldots x_{n-1}) \overset{\mathsf{e}\,g(x_1\ldots x_{n-1})\,x_n}{=\!=\!=\!=\!=\!=} \mathsf{v}(gx_1\ldots x_n)$$

where v (resp. e) is a short notation for vertex (resp. edge).

Proposition 39. *The Cayley complex* $\mathrm{C}\,P$ *is the type* $\overline{Q}$ *generated by the fibered 2-polygraph* Q *with*

$$Q_0 \equiv G \qquad\qquad Q_1 \equiv G \times X \qquad\qquad Q_2 \equiv G \times R$$

and the expected boundary maps.

The fiber sequence of Theorem 35 generalizes to this setting:

Theorem 40. *We have a fiber sequence*

$$\mathrm{C}\,P \longrightarrow \mathrm{B}_2\,P \xrightarrow{\phi} \mathrm{B}\,G$$

Proof. The main arguments of the proof are following ones, which essentially follows the proof of Theorem 35. A more detailed version of this proof (although in the case where G is the quaternion group Q, but not depending on this in an essential way) can be found in [33, Section 3.5].

We write P' for the underlying 1-polygraph of the 2-polygraph P and $\mathrm{B}_1\,P \equiv \overline{P'}$ for the generated type. We have seen in (6.2) that the type $\mathrm{B}_2\,P \equiv \overline{P}$ can be obtained from the type $\mathrm{B}_1\,P$ as the pushout

$$\begin{tikzcd} O \times \Sigma\,\mathrm{S}^1_P\,.P_2 \arrow[r, "p"] \arrow[d, "f"'] & \Sigma\,\mathrm{S}^1_P\,.P_2 \arrow[d, "g"] \\ \mathrm{B}_1\,P \arrow[r, "q"'] & \mathrm{B}_2\,P \arrow[ul, phantom, "\ulcorner", very near start] \end{tikzcd}$$

where the map f essentially sends a 2-generator to the sphere in $\mathrm{B}_1\,P$ corresponding to its boundary and p is the second projection. Now, consider the type family

$$\begin{aligned} F : \mathrm{B}_2\,P &\to \mathcal{U} \\ x &\mapsto (\star = \phi(x)) \end{aligned}$$

By the flattening lemma for pushouts (Section 1), we thus have a pushout of total spaces of the fibration F:

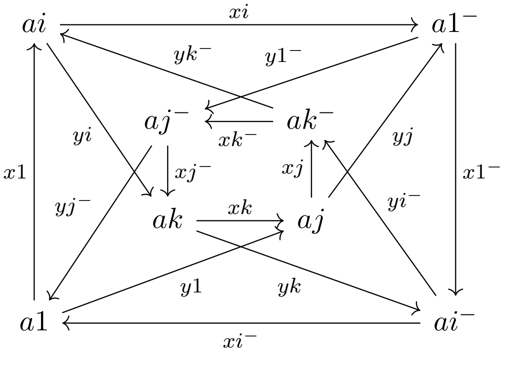
$$\begin{tikzcd} \Sigma(O \times \Sigma\,\mathrm{S}^1_P\,.P_2).F \circ g \circ p \arrow[r, "\Sigma p.(\lambda\_.\,\mathrm{id})"] \arrow[d, "\Sigma f.e"'] & \Sigma(\Sigma\,\mathrm{S}^1_P\,.P_2).F \circ g \arrow[d, "\Sigma g.(\lambda\_.\,\mathrm{id})"] \\ \Sigma(\mathrm{B}_1\,P).F \circ q \arrow[r, "\Sigma q.(\lambda\_.\,\mathrm{id})"'] & \Sigma(\mathrm{B}_2\,P).F \arrow[ul, phantom, very near start, "\ulcorner"] \end{tikzcd}$$

where $e : (x : O \times \Sigma\,\mathrm{S}^1_P\,.P_2) \to F \circ g \circ p(x) \to F \circ q \circ f(x)$ is the map canonically induced by the identity $g \circ p = q \circ f$. By definition, the type $\Sigma(\mathrm{B}_2\,P).F$ on the lower right is the kernel of ϕ. Remembering that $\mathrm{B}_1\,P$ is another notation for $\mathrm{B}\,X^*$, we have by Theorem 35 that the type $\Sigma(\mathrm{B}_1\,P).F \circ q$ at the lower left is the Cayley graph $\mathrm{C}(X, G)$. Namely, an element of $\Sigma(\mathrm{B}_1\,P).F \circ q$ consists of an element of $\mathrm{B}_1\,P$ (i.e. a point in the loop corresponding to an element $x : X$) and an element of G (because $F \circ q(\star)$ is $\star = \star$ in $\mathrm{B}_2\,P$, i.e. G). Now, an element of $\Sigma(O \times \Sigma\,\mathrm{S}^1_P\,.P_2).F \circ g \circ p$ consists in a point $s : O$, a 2-generator of P in some 1-sphere (u, v) together with an element a of G, and the map $\Sigma f.e$ respectively sends the northern and southern hemisphere of O to the paths corresponding to u and v starting from a in $\mathrm{C}(X, G)$, and the pushout states that these should be contracted to a point (which is equivalent to filling in the corresponding sphere in $\mathrm{C}(X, G)$ with a disk [39, Section 6.7]). □

Example 41. Consider the quaternion group

$$Q = \langle e, i, j, k \mid i^2 = e, j^2 = e, k^2 = e, ijk = e, e^2 = 1\rangle$$

It is shown in [33] that its Cayley graph is



and its Cayley complex is shown in Fig. 1 (here, all the squares are filled).

Proposition 42. *The Cayley complex is 1-connected and the map $\phi : \mathrm{B}_2\,P \to \mathrm{B}\,G$ is thus 1-connected.*

Proof. Writing $\star$ for the point corresponding to the neutral element of the group, we have to show that the type $\|\star = x\|_0$ is contractible, for any vertex $x : G$. Since the Cayley complex is connected as a quotient of a connected space (Proposition 36) and being contractible is a proposition, we can suppose that we have a path $\star = x$ and therefore, by path induction, we

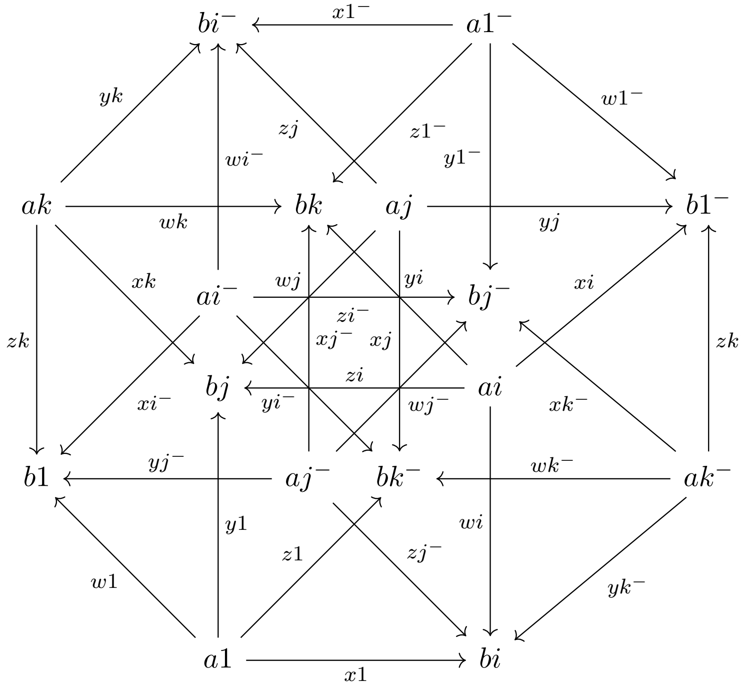


Figure 1: Cayley complex of the quaternion group.

are left with showing that $\|\star = \star\|_0$, i.e. $\pi_1(\mathrm{C}\,P)$ is contractible. By [39, Theorem 8.4.6], the fiber sequence of Theorem 40 induces an exact sequence of homotopy groups

$$\pi_2(\mathrm{B}\,G) \longrightarrow \pi_1(\mathrm{C}\,P) \longrightarrow \pi_1(\mathrm{B}_2\,P) \longrightarrow \pi_1(\mathrm{B}\,G)$$

We have $\pi_2(\mathrm{B}\,G) = 0$ because $\mathrm{B}\,G$ is a groupoid, $\pi_1(\mathrm{B}_2\,P) = G$ by Lemma 37 and $\pi_1(\mathrm{B}\,G) = G$ by definition of deloopings, so that the exact sequence can be simplified to

$$1 \longrightarrow \pi_1(\mathrm{C}\,P) \longrightarrow G \longrightarrow G$$

The arrow on the right, which is $\pi_1(\phi)$, is easily seen to be the identity and the exact sequence states that $\pi_1(\mathrm{C}\,P)$ is the kernel of this map, i.e. 1, and is thus contractible. □

The previous proposition allows us to show that $\mathrm{C}\,P$ can be characterized as the universal covering of $\mathrm{B}_2\,P$. We recall form [18, 32] that a pointed map $f : A \to_* B$ is a *covering* when its fibers are sets. It is a *universal covering* when moreover A is 1-connected.

Proposition 43. *The map $\mathrm{C}\,P \to \mathrm{B}_2\,P$ exhibits $\mathrm{C}\,P$ as the universal covering of $\mathrm{B}_2\,P$.*

Proof. We have seen in Proposition 42 that the type $\mathrm{C}\,P$ is connected, it remains to be shown that its fibers are sets. By [39, Section 8.4], the fiber sequence of Theorem 40 extends on the left as a fiber sequence

$$\Omega\,\mathrm{B}\,G \longrightarrow \mathrm{C}\,P \longrightarrow \mathrm{B}_2\,P$$

Since we have $\Omega\,\mathrm{B}\,G = G$ by definition of $\mathrm{B}\,G$, we deduce that the kernel of the map $\mathrm{C}\,P \to \mathrm{B}_2\,P$ is G and thus a set. Since $\mathrm{B}_2\,P$ is connected and being a set is a proposition [39, Theorem 7.1.10], we deduce that all its fibers are sets. □

In fact, the above situation is very generic. We have observed in (7.1) that the map $\phi : \mathrm{B}_2 P \to \mathrm{B}\, G$ is the groupoid truncation. More generally, given a pointed connected type A, the map $\|-\|_1 : A \to \|A\|_1$ corresponds to the *Galois fibration* associated to A, and its kernel is the universal covering $\tilde{A}$ of A, i.e. we have a fiber sequence

$$\tilde{A} \longrightarrow A \xrightarrow{|-|_1} \|A\|_1$$

which encodes the canonical action of the fundamental group of A on its universal covering. This will be detailed in [32].

Higher Cayley complexes. We expect that the previous constructions extend in higher dimensions. Namely, we should be able to define a notion of *n-polygraph* for every natural number n: essentially, an $(n{+}1)$-polygraph consists in an n-polygraph P together with a function $\mathrm{S}^n_P \to \mathrm{Set}$ which associates to every n-sphere in P a set of *$(n{+}1)$-generators*. The difficulty here lies in the fact that we need to define the notion of n-sphere, which in turn requires defining a notion *n-cell* in P in a coherent way. We leave this for future works. With an $(n{+}1)$-polygraph P, we expect to have a notion of generated type $\overline{P}$ and we say that P is a *presentation* of a group G when $\|\overline{P}\|_n = \mathrm{B}\, G$. Finally, we could define the Cayley complex $\mathrm{C}\, P$ associated to P as the kernel of the truncation map $|-|_n : \overline{P} \to \|\overline{P}\|_n$ so that we have a fiber sequence

$$\mathrm{C}\, P \longrightarrow \overline{P} \longrightarrow \mathrm{B}\, G$$

generalizing Theorems 35 and 40. Note that this would exhibit $\mathrm{C}\, P$ as the universal $(n{-}1)$-covering of $\overline{P}$, in the sense developed in [32]. In particular, Theorems 35 and 40 respectively exhibit the Cayley graph and the Cayley complex as the universal covering and universal 1-covering of the type $\overline{P}$ generated by the presentation P of a given group.

This resolution-like process can be iterated in order to obtain better and better approximations $\mathrm{B}_n\, P$ of $\mathrm{B}\, G$, and higher Cayley complexes as the fibers of the canonical maps $\mathrm{B}_n\, P \to \mathrm{B}\, G$. Moreover, the join construction [35, 36] provides a way to automate this task, see for instance [9, 31].

8. Future works

We have presented two ways to improve the known constructions of deloopings of groups when we have a presentation of the group. This work is part of a larger investigation of models of groups deloopings which are "efficient" in the sense that they allow for using traditional techniques from group presentations, such as the use of Tietze transformations, or coherence based on rewriting [2, 23]. A few instances of this have already been studied in details, but many deloopings of groups remain to be looked at. For instance, a construction of the infinite real projective space was introduced by Buchholtz and Rijke [9] providing a cellular description of $\mathrm{B}\,\mathbb{Z}_2$, we have recently been defining *lens spaces* [31] which provide small cellular deloopings of $\mathrm{B}\,\mathbb{Z}_n$, and we are currently investigating the hypercubical manifolds [33] which provide cellular deloopings for the quaternion group (see also Example 41). More generally, the formalization of group theory in univalent foundations is still under heavy investigation [3], and we aim at developing general techniques to construct efficient representations of (internal) groups in homotopy type theory, which would open the way to cohomological computations for groups [6, 8, 11, 24, 26] or the definition of group actions on higher types (as a generalization of group actions on sets). Finally, we have seen that higher-inductive types play a role in homotopy type theory analogous to the one of polygraphs for strict higher categories. Numerous technique have been developed for those [2], notably based on rewriting, and we would like to adapt them in this setting. The notion of 2-polygraph has been defined in homotopy type theory by Kraus and von Raumer [23] who also explained how to obtain coherence (formally, a homotopy basis) when we have a convergent presentation. In [30], we develop the notion of Tietze transformation for 1-polygraphs and show both their correction and completeness and, in this paper, we have developed a similar notion for 2-polygraphs. A general definition of n-polygraphs is still missing and is the subject of ongoing work. Having a notion of 3-polygraph would already be a great improvement since it would allow for manipulating coherent presentations of groups.

References

[1] The Agda Community. Cubical Agda library, July 2023. URL: `https://github.com/agda/cubical`.

[2] Dimitri Ara, Albert Burroni, Yves Guiraud, Philippe Malbos, François Métayer, and Samuel Mimram. *Polygraphs: From Rewriting to Higher Categories.* London Mathematical Society Lecture Note Series. Cambridge University Press, 2025. `doi:10.1017/9781009498968`.

[3] Marc Bezem, Ulrik Buchholtz, Pierre Cagne, Bjørn Ian Dundas, and Daniel R. Grayson. Symmetry. Commit: `4b08e0e`. URL: `https://github.com/UniMath/SymmetryBook`.

[4] Marc Bezem, Ulrik Buchholtz, Daniel R Grayson, and Michael Shulman. Construction of the circle in UniMath. *Journal of Pure and Applied Algebra*, 225(10):106687, 2021. `doi:10.1016/j.jpaa.2021.106687`.

[5] Guillaume Brunerie. *On the homotopy groups of spheres in homotopy type theory.* PhD thesis, Université de Nice, 2016. `arXiv:1606.05916`.

[6] Guillaume Brunerie, Axel Ljungström, and Anders Mörtberg. Synthetic integral cohomology in cubical Agda. In *30th EACSL Annual Conference on Computer Science Logic (CSL 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022. `doi:10.4230/LIPIcs.CSL.2022.11`.

[7] Ulrik Buchholtz, J. Daniel Christensen, Jarl G. Taxerås Flaten, and Egbert Rijke. Central H-spaces and banded types. *Journal of Pure and Applied Algebra*, 229(6):107963, 2025. `doi:10.1016/j.jpaa.2025.107963`.

[8] Ulrik Buchholtz and Kuen-Bang Hou. Cellular cohomology in homotopy type theory. *Logical Methods in Computer Science*, 16, 2020. `doi:10.23638/LMCS-16(2:7)2020`.

[9] Ulrik Buchholtz and Egbert Rijke. The real projective spaces in homotopy type theory. In *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–8. IEEE, 2017. `doi:10.5555/3329995.3330081`.

[10] Albert Burroni. Higher-dimensional word problems with applications to equational logic. *Theoretical computer science*, 115(1):43–62, 1993. `doi:10.1016/0304-3975(93)90054-W`.

[11] Evan Cavallo. *Synthetic cohomology in homotopy type theory.* PhD thesis, MA thesis, 2015.

[12] Arthur Cayley. Desiderata and suggestions. No. 2 – The theory of groups: graphical representation. *American journal of mathematics*, 1(2):174–176, 1878.

[13] Camil Champin and Samuel Mimram. Delooping generated groups. URL: `https://github.com/smimram/generated-deloopings-agda`.

[14] Camil Champin, Samuel Mimram, and Émile Oleon. Delooping generated groups in homotopy type theory. In Jakob Rehof, editor, *9th International Conference on Formal Structures for Computation and Deduction (FSCD 2024)*, volume 299 of *LIPIcs*, page 6:1–6:20. Schloss Dagstuhl, 2024. `doi:10.4230/LIPIcs.FSCD.2024.6`.

[15] Thierry Coquand and Nils Anders Danielsson. Isomorphism is equality. *Indagationes Mathematicae*, 24(4):1105–1120, 2013. `doi:10.1016/j.indag.2013.09.002`.

[16] Michel Demazure and Pierre Gabriel. *Groupes Algébriques, Tome 1.* North-Holland Publishing Company, 1970.

[17] John R. J. Groves. Rewriting systems and homology of groups. In *Groups—Canberra 1989: Australian National University Group Theory Program 1989*, pages 114–141. Springer, 2006. `doi:10.1007/BFb0100735`.

[18] Robert Harper and Kuen-Bang Hou (Favonia). Covering spaces in homotopy type theory. In *22nd International Conference on Types for Proofs and Programs (TYPES 2016)*, pages 11–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2018. `doi:10.4230/LIPIcs.TYPES.2016.11`.

[19] Krzysztof Kapulkin and Peter LeFanu Lumsdaine. The simplicial model of Univalent Foundations (after Voevodsky). *Journal of the European Mathematical Society*, 23(6):2071–2126, 2021. `doi:10.4171/jems/1050`.

[20] Deepak Kapur and Paliath Narendran. A finite Thue system with decidable word problem and without equivalent finite canonical system. *Theoretical Computer Science*, 35:337–344, 1985. `doi:10.1016/0304-3975(85)90023-4`.

[21] Donald E Knuth and Peter B Bendix. Simple word problems in universal algebras. *Automation of Reasoning: 2: Classical Papers on Computational Logic 1967–1970*, pages 342–376, 1983. `doi:10.1016/B978-0-08-012975-4.50028-X`.

[22] Nicolai Kraus and Thorsten Altenkirch. Free higher groups in homotopy type theory. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 599–608, 2018. `doi:10.1145/3209108.3209183`.

[23] Nicolai Kraus and Jakob von Raumer. A rewriting coherence theorem with applications in homotopy type theory. *Mathematical Structures in Computer Science*, 32(7):982–1014, 2022. `doi:10.1017/S0960129523000026`.

[24] Thomas Lamiaux, Axel Ljungström, and Anders Mörtberg. Computing cohomology rings in cubical Agda. In *Proceedings of the 12th ACM SIGPLAN International Conference on Certified Programs and Proofs*, pages 239–252, 2023. `doi:10.1145/3573105.3575677`.

[25] 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)*, pages 1–9, 2014. `doi:10.1145/2603088.2603153`.

[26] Axel Ljungström and Anders Mörtberg. Computational synthetic cohomology theory in homotopy type theory. Preprint, 2024. `arXiv:2401.16336`.

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

[28] Roger C. Lyndon and Paul E. Schupp. *Combinatorial group theory*, volume 188. Springer, 1977.

[29] Per Martin-Löf. *Intuitionistic type theory*, volume 1 of *Studies in proof theory*. Bibliopolis, 1984.

[30] Samuel Mimram and Émile Oleon. Coherent Tietze transformations of 1-polygraphs in homotopy type theory. Accepted at FSCD 2025.

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

[32] Samuel Mimram and Émile Oleon. The Galois fibration in homotopy type theory. In preparation, 2025.

[33] Samuel Mimram and Émile Oleon. Hypercubical manifolds in homotopy type theory. Submitted, 2025. `arXiv:2506.19402`.

[34] Henri Poincaré. *Analysis situs*. Gauthier-Villars Paris, France, 1895.

[35] Egbert Rijke. The join construction. Unpublished manuscript, 2017. `arXiv:1701.07538`.

[36] Egbert Rijke. *Classifying Types*. PhD thesis, Carnegie Mellon University, July 2018. `arXiv:1906.09435`.

[37] Egbert Rijke. *Introduction to Homotopy Type Theory*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2025.

[38] Heinrich Tietze. Über die topologischen Invarianten mehrdimensionaler Mannigfaltigkeiten. *Monatshefte für Mathematik und Physik*, 19:1–118, 1908.

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

[40] Andrea Vezzosi, Anders Mörtberg, and Andreas Abel. Cubical Agda: A dependently typed programming language with univalence and higher inductive types. *Journal of Functional Programming*, 31, 2021. `doi:10.1145/3341691`.

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

[42] David Wärn. Path spaces of pushouts. Preprint, 2023. `arXiv:2402.12339`.