Delooping generated groups in homotopy type theory

Camil Champin
École Normale Supérieure de Lyon

Samuel Mimram
LIX, CNRS, École polytechnique, Institut Polytechnique de Paris, 91120 Palaiseau, France.

Émile Oleon
LIX, CNRS, École polytechnique, Institut Polytechnique de Paris, 91120 Palaiseau, France.

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 continuous deformation) 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 to construct 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 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. We also investigate, in this context, an abstract construction for the Cayley graph of a generated group and show that it encodes the relations of the group. Most of the developments performed in the article have been formalized using the cubical version of the Agda proof assistant.

2012 ACM Subject Classification Theory of computation → Constructive mathematics

Keywords and phrases homotopy type theory, delooping, group, generator, Cayley graph

Digital Object Identifier 10.4230/LIPIcs.FSCD.2024.3

Supplementary Material *Software (Proofs)*: `https://github.com/smimram/generated-deloopings-agda`

Introduction

Homotopy type theory was introduced around 2010 [27]. It is based on Martin-Löf type theory [20], starting from the idea that types in logic should be interpreted not only as sets, as traditionally done in 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* [14], which states that identities between types coincide with equivalences. This opens the way to the implementation of geometric constructions in a synthetic way, by performing operations on types, which will semantically correspond to the desired operations on spaces. In this setting, we are interested in providing ways to construct models of groups which are concise in order to allow for simple proofs, but also to make the meta-theoretic reasoning easier.

Delooping groups. Following a well-known construction due to Poincaré at the end of the 19th century [22], to any type A which is pointed, i.e. equipped with a distinguished element $\star$, we can associate its *fundamental group* $\pi_1(A) := \|\star = \star\|_0$ whose elements are

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* $\Omega A := (\star = \star)$, defined similarly but without quotienting paths up to homotopy. Once this observation 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 [2, 5], and will be recalled in the article, induces an equivalence between the type of groups and the type of pointed connected groupoids (Theorem 20). 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 traditional one as groups (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 [2]). 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 a particular case of the definition of *Eilenberg-MacLane spaces* in homotopy type theory due to Finster and Licata [17]. It consists in constructing $\mathrm{B}\,G$ as a higher inductive type with one point (so that it is pointed), 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.

The second one is the *torsor* construction which originates in algebraic topology [12] and can be adapted in homotopy type theory [2, 5, 29]. 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.

Smaller deloopings of 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 the definition requires to introduce less material), when a presentation by generators and relation is known for the group.

For the first construction (as a higher inductive type), we show here the 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 2). This has the advantage of resulting in types are simpler to define, require handling less cases when reasoning with those by induction, and are closer to the usual combinatorial description of groups. Moreover, we claim that the traditional methods based on rewriting [13, 1] 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 Raummer's adaptation of Squier's coherence theorem in homotopy type theory [16].

For the second 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-gets by what we call here X-sets (Theorem 11), where we only need to consider the action for 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 [3]: the reason why they use it instead of more traditional one [27] 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 [18], the semantics of type theory [14] 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.

Cayley graphs. As a last 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 [9, 19]. 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 16). This establishes those graphs as a measure of the difference between deloopings and their approximations, and suggests higher dimensional versions of those.

Formalization. Most of the results presented in this article have been formalized in the cubical variant of the Agda proof assistant [28] using the "standard library" which has been developed for it [26]. Our developments are publicly available [10], and we provide pointers to the formalized results.

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 using higher inductive types, and explain how those can be simplified when a presentation is known for the group (Section 3). We then present the other approach for defining delooping of groups based on the torsor construction (Section 4) and show how it can be simplified when a generating set is known for the group (Section 5). Finally, we investigate the construction of Cayley graphs in homotopy type theory (Section 6) and conclude, presenting possible extensions of this work (Section 7).

Acknowledgments. We would like to thank Dan Christensen as well as an anonymous reviewer for useful comments on early drafts of this article.

1 Homotopy type theory

We unfortunately do not have enough space here to provide an introduction from scratch to dependent type theory and homotopy type theory, so we refer the reader to the reference

book for an in depth presentation [27]. The main purpose of this section is to fix some terminology and notations for classical notions.

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 $\Pi(x : A).B$ or $(x : A) \to B$ for Π-types, and $A \to B$ for the case where B is non-dependent. Similarly, we write $\Sigma(x : A).B$ for Σ-types, and $A \times B$ for the non-dependent version. The two projections from a Σ-type are respectively written π and π'.

Paths. Given a type $A : \mathcal{U}$ and two elements $a, b : A$, we write $a =_A b$ 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 $a : A$, the type $a = a$ contains the term refl_a witnessing for reflexivity of equality. We sometimes write $x := t$ to indicate that x and t are equal by definition. The elimination principle of identities, aka *path induction* and often noted J, roughly 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 $x := a$ and $p := \mathrm{refl}_a$. By path induction, the following can be shown. 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 $p^{\to} : A \to B$, 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*.

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 feature a useful generalization of such types, called *higher inductive types*. They allow, in addition to traditional constructors for elements of the type, 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. Higher-dimensional spheres S^n 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. 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 [27, 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 $\Sigma(x : A).(y : A) \to (x = y)$ is inhabited: this means that we have a "contraction point" $a_0 : A$, and a continuous family of paths from a_0 to every other point in A. A type A is a *proposition* (resp. a *set*, resp. a *groupoid*) when $(x = y)$ 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. 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. groupoid).

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. 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 only assume that we have an element of A if B itself is a proposition. 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* $\|A\|_1$.

Fibers. Given a function $f : A \to B$, we write $\mathrm{fib}_f\, b$ for the type $\Sigma(a : A).(f\, a = b)$, called the *fiber* of f at b. The 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.

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. 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 $i(x)$.

The internal point of view. A *pointed* type consists of a type A together with a distinguished element, often written $\star$ and sometimes left implicit. Given a pointed type $(A, \star)$, its *loop space* $\Omega\, A$ is defined as the type of paths from $\star$ to itself: $\Omega\, A := (\star = \star)$. The elements of this type are called *loops*. By path induction one can construct, for every two paths $p : a = b$ and $q : b = c$, a path in $a = c$ called their *concatenation* and written $p \cdot q$. 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 [27, Section 2.1].

Delooping groups. A *delooping* of a group G is a pointed connected groupoid $\mathrm{B}\, G$ together with an identification $\mathrm{d}_G : \Omega\, \mathrm{B}\, G = G$ (we recall that a type A is *connected* when the type $\|A\|_0$ is contractible, i.e. A has one connected component). The notation is justified by the fact that deloopings are unique. 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 $\mathbb{Z}$ [27, Section 8.1].

3 Delooping using higher inductive types

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 [17, Theorem 3.2]. Note that we only ask here that loop preserves multiplication (with loop-comp), because it can be shown that this implies preservation of unit and inverses. In particular, preservation of unit (see `EM.loop-id`) renders superfluous one of the constructors present in the original definition [17].

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. Given a set X, we write X^* for the *free group* over X [27, Theorem 6.11.6]. There is an inclusion function $\iota : X \to X^*$ which, by precomposition, induces an equivalence between morphisms of groups $X^* \to G$ and functions $X \to G$. 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 $a_1 \dots a_n$ 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).

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

$$X \rightrightarrows 1 \dashrightarrow \bigvee_X \mathrm{S}^1 \tag{1}$$

or, equivalently, as the higher inductive type generated by the two constructors $\star : \bigvee_X \mathrm{S}^1$ and $\mathsf{loop} : X \to \star = \star$.

▶ **Proposition 1.** *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 [27, Exercise 8.2] and [15], but the general case is more involved and was recently proved in [30]: 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 $[P] := \|X^* /\!\!/ R\|_0$ obtained by taking the set truncation of the coequalizer

$$R \underset{\pi'}{\overset{\pi}{\rightrightarrows}} X^* \overset{\kappa}{\dashrightarrow} X^* /\!\!/ R$$

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 $P := \langle X \mid R\rangle$, i.e. such that $G = [P]$. We define the type $\mathrm{B}\,P$ as the following higher inductive type:

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

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 1). Our main result in this section is the following:

▶ **Theorem 2** (`EM.theorem`). *Given a presentation $P := \langle X \mid R \rangle$, 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 $f\star := \star$, and $f(\mathsf{gen}\,a) := \mathsf{loop}[a]$ 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 $x := \star$, 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 [27, Example 8.7.17]; the proof suggested there is more involved since it is based on a generalized van Kampen theorem.

▶ **Example 3.** The dihedral group D_5, see Example 13, 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 2 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 4.** 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 2, we actually recover the inductive type $\mathrm{K}(G, 1)$ as delooping of G.

4 Delooping with torsors

In this section, we recall the other classical approach to constructing deloopings of groups by using G-torsors, which originates in classical constructions of algebraic topology [12]. Most of the material of the section is already known, for which reason proofs are not much detailed. A more in-depth presentation can be found in recent works such as [2, 5].

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

$$\alpha(xy) = \alpha(x) \circ \alpha(y) \qquad \alpha(1) = \mathrm{id}_A \tag{2}$$

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

Proof. The type of sets is a groupoid [27, Theorem 7.1.11]. Given a set A, the type of functions $A \to A$ is a set [27, Theorem 7.1.9] and thus a groupoid. Finally, the axioms (2) of actions are propositions (because A is a set) and thus groupoids. We conclude since groupoids are closed under Σ-types [27, 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

$$\beta(x)(f(a)) = f(\alpha(x)(a)). \tag{3}$$

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 6** (`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 [27, Theorem 2.7.2] entails that an equality between $(\mathrm{dom}\,\alpha, \alpha)$ and $(\mathrm{dom}\,\beta, \beta)$ is a pair consisting of an equality $p : \mathrm{dom}\,\alpha = \mathrm{dom}\,\beta$ and an equality $q : p^{\to}(\alpha) = \beta$ (we can forget about the equality between the components expressing the properties required for group actions since those are propositions). By [27, 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 [27, Section 2.14]. In fact, this map from equalities to isomorphisms can itself be shown to be an equivalence:

▶ **Proposition 7** (`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 [11] and [27, Section 9.8], and can be understood as a generalisation of univalence for types having an algebraic structure. ◀

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 8** (`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 $\phi : \operatorname{Aut} \mathrm{P}_G \to G$ and $\psi : G \to \operatorname{Aut} \mathrm{P}_G$, respectively defined by $\phi(f) := f(1)$ and $\psi(x)(y) := yx$ are mutually inverse group morphisms, see Appendix A. ◀

The type Set_G is thus "almost" a delooping of G. Namely, it is a groupoid (Lemma 5), which is pointed by P_G and satisfies $\Omega\,\mathrm{Set}_G = G$ by Propositions 7 and 8. It only lacks being connected, which can easily be addressed. Given a pointed type A, its *connected component* $\operatorname{Comp} A$ is the type $\Sigma(x : A).\|\star = x\|_{-1}$. It is well-known that this type is pointed by $(\star, |\operatorname{refl}|_{-1})$, connected and has the same loop space as the original type, i.e. we have $\Omega \operatorname{Comp} A = \Omega A$, see Appendix C. We thus have:

▶ **Theorem 9** (`Deloopings.torsorDeloops`). *The connected component of P_G in Set_G, i.e. the type $\operatorname{Comp}(\mathrm{Set}_G, \mathrm{P}_G)$, is a delooping of G.*

The elements of the connected component of the principal G-set are usually called *G-torsors*.

5 Generated torsors

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 3, 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

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

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) for every $x : X$. The identities between X-sets

can be characterized in a similar way as for G-sets, see Proposition 6, and Proposition 7 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 11 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 9, 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 is 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 $\mathrm{Set}^{\circlearrowleft} := \Sigma(A : \mathrm{Set}).\, \mathrm{End}\, A$ for the type of all *endomorphisms* (on any set), our theorem will imply that the type

$$\Sigma((A, f) : \mathrm{Set}^{\circlearrowleft}).\|(\mathbb{Z}_n, s) = (A, f)\|_{-1} \tag{4}$$

(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 [3], 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 [8, 6, 4]. 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 10** (`XSetProperties.theorem`). *We have a group equivalence* $\Omega\,\mathrm{P}_G \simeq \Omega\,\mathrm{P}_X$.

Proof. From Proposition 6, 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

$$g(p^{\rightarrow}(z)) = p^{\rightarrow}(gz) \tag{5}$$

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

$$\gamma(x)(p^{\rightarrow}(z)) = p^{\rightarrow}(\gamma(x)z) \tag{6}$$

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 (6) is a particular case of (5). 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 (6) 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 (5) 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 (5) is a proposition, by the elimination principle of propositional truncation, we can actually suppose given such a u, and we have

$$\begin{aligned} x(p^{\rightarrow}(y)) &= \gamma^*(u)(p^{\rightarrow}(y)) && \text{since } \gamma^*(u) = x \\ &= p^{\rightarrow}(\gamma^*(u)y) && \text{by repeated application of (6)} \\ &= p^{\rightarrow}(xy) && \text{since } \gamma^*(u) = x. \end{aligned}$$

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

$$\begin{array}{ccccccccc} G & \xrightarrow{\gamma(x_1)} & G & \xrightarrow{\gamma(x_2)} & G & \longrightarrow \dots \longrightarrow & G & \xrightarrow{\gamma(x_n)} & G \\ {\scriptstyle p^{\rightarrow}}\downarrow & & {\scriptstyle p^{\rightarrow}}\downarrow & & {\scriptstyle p^{\rightarrow}}\downarrow & & \downarrow{\scriptstyle p^{\rightarrow}} & & \downarrow{\scriptstyle p^{\rightarrow}} \\ G & \xrightarrow[\gamma(x_1)]{} & G & \xrightarrow[\gamma(x_2)]{} & G & \longrightarrow \dots \longrightarrow & G & \xrightarrow[\gamma(x_n)]{} & G \end{array}$$

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

Proof. Since taking the connected component preserves loops spaces (Proposition 23), 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 10, and thus to G by Theorem 9. ◀

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 [27, 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 12.** *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) of $\mathbb{Z}_n$ can slightly be simplified as

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

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

▶ **Example 13.** Theorem 11 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.

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

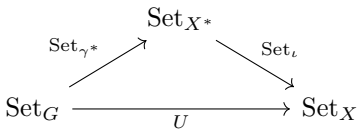
is a delooping of the dihedral group D_n.

Alternative proof. We would like to provide another proof Proposition 10, 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 [27, Definition 4.6.1]. It relies on the following result whose proof can be found in Appendix A.

▶ **Lemma 14.** *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 $a : A$.*

▶ **Proposition 15.** *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 $U := \mathrm{Set}_\gamma$. The function $\gamma : X \to G$ can be decomposed as $\gamma^* \circ \iota$, and therefore Set_γ can be decomposed as $\mathrm{Set}_\iota \circ \mathrm{Set}_{\gamma^*}$:



Both maps are embeddings so that U is an embedding by composition. Namely,

- Set_{γ^*} *is an embedding.* Consider the map

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

  obtained by precomposition by γ^*. Since γ^* is surjective we have that $F\,A$ is an embedding for every set A [27, Lemma 10.1.4]. By Lemma 14, we deduce that $\Sigma\,\mathrm{Set}\,.F$, which is Set_{γ^*}, is an embedding.
- Set_ι *is an embedding.* By universal property of X^*, given a set A, the map

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

  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

  $$(X \to \mathrm{Aut}\, A) \to (X \to \mathrm{End}\, A)$$

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

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

  is such that $F\,A$ is an embedding for every set A. By Lemma 14, the map $\Sigma\,\mathrm{Set}\,.F$, which is Set_ι, is thus an embedding. ◀

6 Cayley graphs

We have seen in Section 3 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 [9, 19], 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

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

For instance, the Cayley graphs associated to $\mathbb{Z}_5$ (with 2 as generator) and D_5 (with r and s as generators) are respectively

Our main result in this section is that this type satisfies the following property. We recall that the delooping operation is functorial, see Appendix B and [29]: in particular, we can deloop morphisms. We also recall that the *kernel* of a map $f : A \to B$ with B pointed is its fiber at the distinguished element $\star$ of B.

▶ **Theorem 16** (`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 define the type family $P : \mathrm{B}\,X^* \to \mathcal{U}$ by $P(x) = (\star = \mathrm{B}\,\gamma^*(x))$. Remember that $\mathrm{B}\,X^*$ admits a description as a coequalizer, see (1) and Proposition 1. Hence, by the flattening lemma for coequalizers (see Lemma 25 and [27, Section 6.12]), we have a coequalizer of total spaces

$$\Sigma X.P(\star) \underset{(x,p)\mapsto(x\cdot\star,\,P_p^{\rightarrow}(\star))}{\overset{(x,p)\mapsto(\star,p)}{\rightrightarrows}} \Sigma 1.P(\star) \dashrightarrow \Sigma\,\mathrm{B}\,X^*.P$$

By using the properties of transport in path spaces [27, 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 $P(\star)$ is equal to $\Omega\,\mathrm{B}\,G$, i.e. G, and we have the following coequalizer:

$$X \times G \underset{(x,g)\mapsto gx}{\overset{(x,g)\mapsto g}{\rightrightarrows}} G \dashrightarrow \Sigma\,\mathrm{B}\,X^*.P$$

If follows that $\Sigma\,\mathrm{B}\,X^*.P$ 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 (see [27, Section 8.4])

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

which encodes the fact that we have an action of G on its Cayley graph, whose homotopy quotient $\mathrm{C}(X, G)/\!\!/G$ is $\mathrm{B}\, X^*$, see [21, Proposition 16]. Hence, we recover the canonical action of G on its Cayley graph.

Relations. The long exact sequence of homotopy groups induced by the above fiber sequence [27, Theorem 8.4.6] implies in particular that we have the following short exact sequence of groups

$$0 \longrightarrow \Omega\, \mathrm{C}(X, G) \longrightarrow X^* \longrightarrow G \longrightarrow 0$$

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 16 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 3.

The Cayley complex and higher variants. We now briefly explain that we can extend the previous construction in higher dimensions in order to define internally a type corresponding to the classical *Cayley complex* [19]. Suppose given a presentation $P := \langle X \mid R \rangle$ for G and write $\mathrm{B}_2\, P$ for the 2-skeleton of the type $\mathrm{B}\, P$ defined in Section 3 (i.e. the type generated by $\star$, gen, and rel, but without the truncation trunc). As in previous section, this type can be considered as an approximation of $\mathrm{B}\, P$ (lacking the truncation) and we would like to measure the difference between the two types. The *Cayley complex* $\mathrm{C}\, P$ associated to the presentation P is the inductive type defined by

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

where, given $g : G$ and $u := x_1 x_2 \dots x_n : X^*$, we have that $(\mathsf{edge}\, g)^*\, u$ is the path

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

where v (resp. e) is a short notation for vertex (resp. edge). We will detail in future works the proof of the following result, which can be performed using the flattening lemma for pushouts, as in Theorem 16:

▶ **Theorem 17.** *We have a fiber sequence* $\mathrm{C}\, P \longrightarrow \mathrm{B}_2\, P \longrightarrow \mathrm{B}\, G$.

In fact, 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 [23, 24] provides a way to automate this task, see for instance [7, 21].

7 Future works

We have presented two ways to improve in practice the known constructions of deloopings of groups when we have a presentation of the group. This work is part of a larger investigation of "efficient" models of groups deloopings, in the sense that we can compute effectively with those. In particular, the construction of the infinite real projective space performed by Buchholtz and Rijke [7], provides a cellular description of $\mathrm{B}\, \mathbb{Z}_2$ (which is better than the usual ones obtained by generic methods because it consists in a *non-recursive* higher inductive type). In other current work, we refine their approach in order to construct lens spaces and thus obtain a cellular version of $\mathrm{B}\, \mathbb{Z}_n$ for every natural number n, as well as efficient

representations of deloopings of other classical groups [21]. More generally, the formalization of group theory in univalent foundations is still under heavy investigation [2], 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 [8, 6, 4] or the definition of group actions on higher types (as a generalization of group actions on sets). On another note, higher-inductive types play a role in homotopy type theory analogous to the one of polygraphs for strict higher categories: numerous techniques have been developed for those [1], notably based on rewriting, and we plan to adapt them in this setting, following first developments of [16].

References

1 Dimitri Ara, Albert Burroni, Yves Guiraud, Philippe Malbos, François Métayer, and Samuel Mimram. Polygraphs: From rewriting to higher categories. To appear, 2023. arXiv:2312.00429.

2 Marc Bezem, Ulrik Buchholtz, Pierre Cagne, Bjørn Ian Dundas, and Daniel R. Grayson. Symmetry. URL: https://github.com/UniMath/SymmetryBook.

3 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. arXiv:1910.01856, doi:10.1016/j.jpaa.2021.106687.

4 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.

5 Ulrik Buchholtz, J Daniel Christensen, Jarl G Taxerås Flaten, and Egbert Rijke. Central H-spaces and banded types. Preprint, 2023. arXiv:2301.02636.

6 Ulrik Buchholtz and Kuen-Bang Hou. Cellular cohomology in homotopy type theory. *Logical Methods in Computer Science*, 16, 2020. arXiv:1802.02191, doi:10.23638/LMCS-16(2:7)2020.

7 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. arXiv:1704.05770, doi:10.5555/3329995.3330081.

8 Evan Cavallo. *Synthetic cohomology in homotopy type theory*. PhD thesis, MA thesis, 2015.

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

10 Camil Champin and Samuel Mimram. Delooping generated groups. URL: https://github.com/smimram/generated-deloopings-agda.

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

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

13 John RJ 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.

14 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. arXiv:1211.2851, doi:10.4171/jems/1050.

15 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. arXiv:1805.02069, doi:10.1145/3209108.3209183.

16 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. arXiv:2107.01594, doi:10.1017/S0960129523000026.

17 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.

18 Peter LeFanu Lumsdaine and Michael Shulman. Semantics of higher inductive types. *Mathematical Proceedings of the Cambridge Philosophical Society*, 169(1):159–208, 2020. arXiv:1705.07088, doi:10.1017/S030500411900015X.

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

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

21 Émile Oleon and Samuel Mimram. Delooping cyclic groups with lens spaces in homotopy type theory. Accepted at LICS 2024 conference, 2024.

22 Henri Poincaré. *Analysis situs*. Gauthier-Villars Paris, France, 1895.

23 Egbert Rijke. The join construction. Preprint, 2017. arXiv:1701.07538.

24 Egbert Rijke. *Classifying Types*. PhD thesis, Carnegie Mellon University, July 2018. arXiv:1906.09435.

25 Egbert Rijke. Introduction to homotopy type theory. Preprint, 2022. arXiv:2212.11082.

26 The Agda Community. Cubical Agda Library, July 2023. URL: https://github.com/agda/cubical.

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

28 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.

29 David Wärn. Eilenberg–maclane spaces and stabilisation in homotopy type theory. *Journal of Homotopy and Related Structures*, 18(2):357–368, 2023. arXiv:2301.03685, doi:10.1007/s40062-023-00330-5.

30 David Wärn. Path spaces of pushouts. Preprint, 2023. arXiv:2402.12339.

A Omitted proofs

Proof of Proposition 8. The two functions

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

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. ◀

Proof of Lemma 14. By definition, the map $\Sigma A.f$ is an embedding iff for every (a, x) and (a', x') in ΣAX, 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 [27, 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 [27, 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 $a : A$. ◀

B Equivalence between internal and external groups

Functoriality of delooping. One of the main properties of the delooping operation is that it is a "local inverse" to taking loop spaces in the following sense:

▶ **Proposition 18.** *Given pointed connected groupoids A and B and a group morphism $f : \Omega\, A \to \Omega\, B$, there is a unique pointed morphism $g : A \to B$ such that $\Omega\, g = f$.*

Proof. By [29, Corollary 12], with $n = 0$, we have that the type $\Sigma(g : A \to B).\, \Omega\, g = f$ is equivalent to $(p, q : \star_A = \star_A) \to f(p \cdot q) = f(p) \cdot f(q)$. This type is a proposition because $\Omega\, B$ is a set (because B is a groupoid), and inhabited (because f is a morphism of groups), and thus contractible. ◀

As an immediate consequence of the above lemma, deloopings are unique:

▶ **Proposition 19.** *Given two deloopings $\mathrm{B}\, G$ and $\mathrm{B}'\, G$ of a group G, we have $\mathrm{B}\, G = \mathrm{B}'\, G$.*

Given a group morphism $f : G \to H$ such that both G and H admit deloopings (and this actually always holds by Theorem 9), the *delooping* of f is the morphism

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

associated, by Proposition 18, to the morphism $\mathrm{d}_H^{\leftarrow} \circ f \circ \mathrm{d}_G^{\rightarrow} : \Omega\, \mathrm{B}\, G \to \Omega\, \mathrm{B}\, H$. By Proposition 18, 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 [2]. We write IntGroup for the type of internal groups, i.e. pointed connected groupoids.

▶ **Theorem 20.** *The maps* Ω : IntGroup $\to$ Group *and* B : Group $\to$ 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 19. ◀

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. 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* [25, Axiom 18.1.8].

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 [27, Theorem 7.1.11], by Theorem 20, 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 show 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.

C Connected components

We define the *connected component* of the pointed type A as the type of points which are merely connected to the distinguished point of A. This type is noted $\mathrm{Comp}\, A$ (or $\mathrm{Comp}(A, \star)$ when we want to specify the distinguished element $\star$). Formally,

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

This type is canonically pointed by $(\star, |\,\mathrm{refl}\,|_{-1})$. 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 21** (`Comp.isConnectedComp`). *The type* $\mathrm{Comp}\, A$ *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 [27, 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 22** (`Comp.loopCompIsLoop`). *We have* $\Omega \operatorname{Comp} A = \Omega A$.

Proof. We begin by showing that the type

$$\Sigma((x, t) : \operatorname{Comp} A).(\star = x) \tag{7}$$

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 [27, 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 (7). We write $\star'$ for the element $(\star, \|\operatorname{refl}_\star\|_{-1})$ of $\operatorname{Comp} A$. The type (7) is pointed by the canonical element $(\star', \operatorname{refl})$ and thus contractible as a pointed proposition.

We have a morphism

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

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

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

between the corresponding total spaces. Since the left member is contractible (by [27, Lemma 3.11.8] again) and the right member is contractible (as shown above), this is an equivalence. By [27, Theorem 4.7.7], for every $x : \operatorname{Comp} A$, the fiber morphism Fx 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 23.** *Given a pointed groupoid* A, $\operatorname{Comp} A$ *is a delooping of* ΩA.

▶ Remark 24. Some people write $\operatorname{Aut} A$ for ΩA and the above proposition states that we have $\operatorname{B} \operatorname{Aut} A = \operatorname{Comp} A$. For this reason, the (confusing) notation $\operatorname{BAut} A$ is also found in the literature for $\operatorname{Comp} A$.

D The flattening lemma

We recall here the classical flattening lemma, see [27, Section 6.12] for a more detailed presentation and proof.

▶ **Lemma 25** (Flattening for coequalizers). *Suppose given a coequalizer*

$$A \underset{g}{\overset{f}{\rightrightarrows}} B \overset{h}{\dashrightarrow} C$$

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

$$\Sigma A.(P \circ h \circ f) \underset{\Sigma g.e}{\overset{\Sigma f.(\lambda\_.\,\mathrm{id}\,)}{\rightrightarrows}} \Sigma B.(P \circ h) \overset{\Sigma h.(\lambda\_.\,\mathrm{id}\,)}{\dashrightarrow} \Sigma C.P$$

is a coequalizer, where the map

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

is induced by transport along p by

$$e\, a\, x := P^{\to}_{p(a)}(x).$$

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.