Data Types with Symmetries via Action Containers

Philipp Joram
Department of Software Science, Tallinn University of Technology, Estonia

Niccolò Veltri
Department of Software Science, Tallinn University of Technology, Estonia

Abstract

We study two kinds of containers for data types with symmetries in homotopy type theory, and clarify their relationship by introducing the intermediate notion of action containers. Quotient containers are set-valued containers with groups of permissible permutations of positions, interpreted as (possibly non-finitary) analytic functors on the category of sets. Symmetric containers encode symmetries in a groupoid of shapes, and are interpreted accordingly as polynomial functors on the 2-category of groupoids.

Action containers are endowed with groups that act on their positions, with morphisms preserving the actions. We show that, as a category, action containers are equivalent to the free coproduct completion of a category of group actions. We derive that they model non-inductive single-variable strictly positive types in the sense of Abbott et al.: The category of action containers is closed under arbitrary (co)products and exponentiation with constants. We equip this category with the structure of a locally groupoidal 2-category, and prove that it locally embeds into the 2-category of symmetric containers. This follows from the embedding of a 2-category of groups into the 2-category of groupoids, extending the delooping construction.

2012 ACM Subject Classification Theory of computation → Type theory; Theory of computation → Categorical semantics

Keywords and phrases Containers, Homotopy Type Theory, Agda, 2-categories

Digital Object Identifier 10.4230/LIPIcs.TYPES.2024.6

Supplementary Material *Software (Agda)*: `https://github.com/phijor/cubical-containers`
archived at `swh:1:dir:a77c8b642661074b9804670b14cf97cc844aff16`

Funding This work was supported by the Estonian Research Council grant PSG749.
Philipp Joram: Participation at TYPES'24 was supported by the COST Action CA20111 – European Research Network on Formal Proofs (EuroProofNet)

1 Introduction

Containers are a representation of strictly positive data types, introduced by Abbott et al. [1]. A container consists of a type of *shapes* and a type of *positions* associated to each shape. For example, the container of ordered lists consists of natural numbers $n : \mathbb{N}$ as shapes and finite ordinals $\mathsf{Fin}(n)$ as positions, the idea being that each list has a length n and $\mathsf{Fin}(n)$-many locations that can hold data. Containers can be interpreted as polynomial endofunctors, the latter being the polymorphic data types that the containers represent. For example, the interpretation of the list container is the List functor.

Containers form a category. Its morphisms represent polymorphic functions between data types, which are interpreted as natural transformations between the corresponding polynomial endofunctors. This category is rich in structure; among other things it is cocartesian closed, is closed under the construction of initial algebras and terminal coalgebras, and admits a notion of derivative.

Traditionally, the theory of containers is studied in Set-like categories. When interpreted in such categories, the data of containers may however be too restrictive to encode certain data types of interest. This is especially the case if one wants to account for symmetries, i.e. identify configurations of positions when one can turn into the other via the action of certain permutations. For example, one can represent ordered lists, but it is not possible to represent cyclic lists or finite multisets as a container.

Some efforts have been made to enhance the expressivity of containers to represent data types with symmetries. Abbott et al. [3] introduced quotient containers, which are containers in which the assignment of values to positions is invariant under a collection of permutations on positions. This is realized by requiring the presence of a subgroup of the symmetric group on positions, corresponding to the collection of admissible permutations. Interpretation of quotient containers embeds them as a subcategory of set-endofunctors, targeting certain quotients of polynomial endofunctors. For quotient containers with finitary positions, these correspond to Joyal's analytic functors [15].

Symmetric containers, introduced by Gylterud [10], consist of a groupoid of shapes, with positions being a set-valued functor over this groupoid. This means that the symmetries are encoded directly in the isomorphisms of the shape groupoid. From the perspective of (homotopy) type theory, symmetric containers correspond to set-bundles over homotopy-groupoids. They form a locally univalent 2-category, and are interpreted as polynomial endofunctors on the 2-category of groupoids.

Quotient and symmetric containers are two different ways to extend the expressivity of ordinary containers to include symmetries between positions. Our interest lies in understanding how these two approaches are related. To this end, we introduce an intermediate notion: *action containers*. An action container consists of a set of shapes and, for each shape s, a set of positions P_s and a group G_s acting on P_s. On one side, such containers generalize quotient containers, as the allowed permutations are determined by the action of an arbitrary group, and are not restricted to subgroups of the symmetric groups. On the other, they are a special case of symmetric containers: a G_s-action is a functor from G_s (as a 1-object groupoid) to Set, and summation of these functors over all shapes s yields a symmetric container.

We describe a notion of morphisms of action containers, inspired by (pre)morphisms of quotient containers. Differently from the latter, morphisms of action containers have to explicitly preserve the structure of the groups acting on positions. Action containers form a category, and we show that this category is the free coproduct completion of a category of group actions in the form of the Fam-construction. From this we derive closure under arbitrary products and coproducts, as well as exponentials with constant domain.

To compare action containers with the 2-category of symmetric containers, we define a notion of invertible 2-cell between these morphisms, enriching them to a 2-category. Crucially, we observe that this 2-category can again be modularly constructed starting from a 2-category of groups, group homomorphisms and *conjugators* [12] using the techniques of displayed bicategories [6]: we first define a 2-category of group actions, displayed over this 2-category of groups, then repeat a 2-categorical version of the Fam-construction, presenting the 2-category of action containers as that of families of group actions.

We construct a 2-functor between the 2-categories of action containers and symmetric containers; again in multiple steps. First we notice that the delooping of a group extends to a 2-functor $\mathbf{B} : \mathsf{Group} \to \mathsf{hGpd}$ between the 2-categories of groups and groupoids, and that this is locally a weak equivalence. We show, again using the displayed machinery, how to lift this to a local weak equivalence $\bar{\mathbf{B}} : \mathsf{Action} \to \mathsf{SetBundle}$ between the 2-categories of group actions and set bundles.

The Fam-construction yields a 2-functor $\mathrm{Fam}(\bar{\mathbf{B}}) : \mathrm{Fam}(\mathsf{Action}) \to \mathrm{Fam}(\mathsf{SetBundle})$ between the 2-categories of families of group actions, i.e. action containers, and families of set bundles. We show that the action of Fam preserves local fully-faithfullness, but that preservation of local essential surjectivity requires an application of the axiom of choice. Finally, we describe a 2-functor $\Sigma : \mathrm{Fam}(\mathsf{SetBundle}) \to \mathsf{SetBundle}$ performing *summations* of family of set bundles, implicitly employing the universal property of the Fam-construction as free coproduct completion. The latter does not seem to be fully faithful, however its restriction to set bundles with connected bases is. This implies that the composite 2-functor $\Sigma \circ \mathrm{Fam}(\bar{\mathbf{B}}) : \mathrm{Fam}(\mathsf{Action}) \to \mathsf{SetBundle}$ is locally fully faithful. This means that, not only do action containers correspond to certain symmetric containers, but so do their morphisms: conjugators between action container morphisms represent exactly identifications of symmetric container morphisms.

We implement our results using the Agda proof assistant, building on top of the `agda/cubical` [19] and `cubical-categorical-logic` [17] libraries. Our code is freely available at `https://github.com/phijor/cubical-containers`. The repository contains a `README` linking results of this paper to the corresponding formalization in the code.

1.1 Notation and Background

Our work takes place in homotopy type theory, which is a well-suited foundational framework for our investigation. This is mostly due to the fact that we can work with syntethic groupoids, i.e. h-groupoids, in place of categorical groupoids, which considerably simplify the described constructions. We take full advantage of higher inductive types (HITs) to define mere existence via propositional truncation, set quotients and the delooping of groups. In this short section, we recall some basic terminology and fix the notation we use. More details can be found in the HoTT book [20].

We write $\prod_{a:A} B(a)$ for dependent products, $A \to B$ for their non-dependent variant, and $\sum_{a:A} B(a)$ for dependent sums. Two-argument function application is written $f(a, b)$ or, to reduce visual clutter, $f_a(b)$. Most of our constructions are universe-polymorphic, but for the sake of readability in the paper we use only the lowest universe of types, denoted $\mathcal{U}$. The type of h-sets is hSet, the type of h-groupoids is hGpd. We will often simply talk about sets and groupoids instead of h-sets and h-groupoids. We suppress proofs of truncation level when they are routine, e.g. for nested Σ- and Π-types. The type of natural numbers is $\mathbb{N}$ and the finite ordinals $\mathsf{Fin} : \mathbb{N} \to \mathsf{hSet}$. For $x, y : A$, we denote their type of identifications by $x = y$, and call $p : x = y$ either an identification or a path. Given a family B over A and terms $x' : B(x)$, $y' : B(y)$, we write $x' =_{B(p)} y'$ for the type of dependent paths, or $x' =_p y'$ when B can be inferred. Defining equalities are denoted $x := y$; for judgmentally equal x and y, we write $x \doteq y$. For functions into Σ-types, we use binders to name the projections: given $f : X \to \sum_{a:A} B(a)$, we write $\lambda x.\,(a(x), b(x))$ or $\lambda x.\,(a_x, b_x)$ for $a := \mathsf{fst} \circ f$ and $b := \mathsf{snd} \circ f$.

The propositional truncation of a type X is $\|X\|_{-1}$, the set truncation of X is $\|X\|_0$. Notice that there are two competing conventions for indexing truncation levels: (-2)-based (HoTT-book style) and 0-based (Voevodsky-style). Our formalization, done in Cubical Agda, is 0-based, yet this paper, which is written in HoTT-book style, starts indexing at -2. Whenever possible however, we will explicitly use the words "proposition", "set" and "groupoid" to avoid confusion. Mere existential quantification is defined as $\exists_{a:A} B(a) := \|\sum_{a:A} B(a)\|_{-1}$. The set quotient of a type A by a (possibly non-propositional) relation R is A/R. The circle S^1 is a HIT with constructors $\bullet : S^1$ and $\mathsf{loop} : \bullet = \bullet$. Propositional and set truncations, as well as set quotients and the circle, are defined as higher inductive types. The main HIT employed in this paper is the delooping of a group, introduced in Section 2.3.

Given a pointed type (X, x_0), its loop space is $\Omega(X, x_0) := (x_0 = x_0)$, and its fundamental group is its set truncation $\pi_1(X, x_0) := \|\Omega(X, x_0)\|_0$. The connected components of a type X are collected in its set truncation $\pi_0(X) := \|X\|_0$. We say that X is connected if $\|X\|_0$ is contractible.

For groups G and H we denote the type of group homomorphisms by $G \mathrel{\dot{\to}} H$. We denote the type of subgroup inclusions by $G \leq H := \sum(\iota : G \mathrel{\dot{\to}} H).\, \mathsf{isInjective}(\iota)$. The symmetric group of a set X is $\mathfrak{S}(X) := \Omega(\mathsf{hSet}, X)$. The unit of this group is reflexivity refl, multiplication is composition of paths $p \cdot q$, inverse is path reversal. Recall that, by univalence, $\mathfrak{S}(X)$ is equivalent to the type of equivalences $X \simeq X$. We abbreviate $\mathfrak{S}(n) := \mathfrak{S}(\mathsf{Fin}(n))$. An action of G on a set X is a group homomorphism $\sigma : G \mathrel{\dot{\to}} \mathfrak{S}(X)$. For $g : G$, we denote $\mathsf{transport}(\sigma(g)) : X \to X$ simply by $\sigma(g)$, and apply it to some $x : X$ either as $\sigma(g, x)$ or $\sigma_g(x)$. The action is said to be faithful if σ is injective. The set of σ-orbits is denoted X/G, and defined as the set quotient of X by the orbit relation $x \sim y := \exists g : G.\, x = \sigma_g(y)$.

1.2 2-Categories

In this paper, we make use of higher categories in the form of (2,1)-categories. We follow the definitions of bicategorical concepts of [13], and adapt them to the setting of homotopy type theory: a 2-category C consist of a type of objects $x, y : \mathsf{C}_0$, 1-cells $f, g : \mathsf{C}_1(x, y)$, and 2-cells $r, s : \mathsf{C}_2(f, g)$, with horizontal composition of 1- and 2-cells, and vertical composition of 2-cells, subject to suitable axioms. In particular, all types of 2-cells $\mathsf{C}_2(f, g)$ are assumed to be *h*-sets. Composition of 1-cells is unital and associative up to a chosen identification, not just a 2-cell. All instances of 2-categories considered here are either locally strict (i.e. 1-cells form sets) or locally univalent; such 2-categories always admit a unique coherently strict structure.

If C is understood from context, we write $f, g : x \to y$ for $f, g : \mathsf{C}_1(x, y)$, and $r : f \Rightarrow g$ for $r : \mathsf{C}_2(f, g)$. We compose cells in *diagrammatic* order. Juxtaposition denotes horizontal composition, whereas vertical composition of 2-cells is denoted $r \bullet s$.

Let $f, g : x \to y$. Under vertical composition, 2-cells $\mathsf{C}_2(f, g)$ form the morphisms of an (ordinary) category, called the *local category* at x and y, denoted by its type of objects $\mathsf{C}_1(x, y)$. If a proposition P holds for all local categories of a 2-category, we say that it is *locally P*. A *(2,1)-category* is thus defined to be a locally groupoidal 2-category, that is, one for which 2-cells in each local category are invertible. A 2-category is locally *thin* if $\mathsf{C}_2(f, g)$ is a proposition for each pair of 1-cells $f, g : \mathsf{C}_1(x, y)$, i.e. there is at most one 2-cell from f to g. Any ordinary category $\mathcal{C}$ forms a locally thin 2-category: 2-cells are homotopies of 1-cells, $\mathcal{C}_2(f, g) := (f = g)$.

We use the machinery of *displayed bicategories* [6] to define complex 2-categories from modular gadgets. A *displayed 2-category* D over a base 2-category C consists of a family of objects $\mathsf{D}_0 : \mathsf{C}_0 \to \mathcal{U}$, a family of 1-cells $\mathsf{D}_1 : \mathsf{C}_1(x, y) \to \mathsf{D}_0(x) \to \mathsf{D}_0(y) \to \mathcal{U}$, and a family of 2-cells $\mathsf{D}_2 : \mathsf{C}_2(f, g) \to \mathsf{D}_1(f; \bar{x}, \bar{y}) \to \mathsf{D}_1(g; \bar{x}, \bar{y}) \to \mathcal{U}$, satisfying dependent analogues of the 2-category axioms. If unambiguous, we write $\bar{f} : \bar{x} \to_f \bar{y}$ for $\bar{f} : \mathsf{D}_1(f; \bar{x}, \bar{y})$, as well as $\bar{r} : \bar{f} \Rightarrow_r \bar{g}$ for $\bar{r} : \mathsf{D}_2(r; \bar{f}, \bar{g})$. The *total 2-category* of D over C is denoted $\int \mathsf{D}$, and is a 2-categorical analouge of $\sum$-types: objects are pairs of objects $\int_0 \mathsf{D} := \sum_{x:\mathsf{C}_0} \mathsf{D}_0(x)$, 1-cells are $\int_1 \mathsf{D}\big((x, \bar{x}), (y, \bar{y})\big) := \sum_{f:\mathsf{C}_1(x,y)} \mathsf{D}_1(f; \bar{y}, \bar{x})$, and 2-cells $\int_2 \mathsf{D}\big((f, \bar{f}), (g, \bar{g})\big) := \sum_{r:\mathsf{C}_2(f,g)} \mathsf{D}_2(r; \bar{f}, \bar{g})$. To highlight the dependency on C, we sometimes write $\int_{\mathsf{C}} \mathsf{D}$ or even $\int_{x:\mathsf{C}} \mathsf{D}(x)$.

We go between 2-categories via 2-functors. For a 2-functor $F : \mathsf{C} \to \mathsf{D}$ we denote its action on objects, 1-, and 2-cells by F_0, F_1 and F_2 respectively. They are assumed to strictly preserve composition and units of 1-cells, that is up to an identification of 1-cells in the codomain. The actions on 1- and 2-cells define functors of local categories, and we call F *locally P* if all local functors satisfy a proposition P.

We define a notion of *displayed 2-functor* $\bar{F} : \bar{\mathsf{C}} \to_F \bar{\mathsf{D}}$ between 2-categories displayed over C and D and a (base) 2-functor $F : \mathsf{C} \to \mathsf{D}$. To cells in $\bar{\mathsf{C}}$ it assigns cells in $\bar{\mathsf{D}}$ displayed over the image of F: it consists of assignments of objects $\bar{F}_0 : \bar{\mathsf{C}}_0(x) \to \bar{\mathsf{D}}_0(F_0(x))$, 1-cells

$\bar{F}_1 : \bar{\mathsf{C}}_1(f; \bar{x}, \bar{y}) \to \bar{\mathsf{D}}_1(F_1(f); \bar{F}_0(\bar{x}), \bar{F}_0(\bar{y}))$ and 2-cells $\bar{F}_2 : \bar{\mathsf{C}}_2(r; \bar{f}, \bar{g}) \to \bar{\mathsf{D}}_2(F_2 r; \bar{F}_1 \bar{f}, \bar{F}_1 \bar{g})$, satisfying dependent analogues of the 2-functor axioms. A displayed 2-functor induces a 2-functor between total 2-categories, denoted $\int \bar{F} : \int \mathsf{C} \to \int \mathsf{D}$.

The 2-category of h-groupoids is denoted by hGpd. Its 1-cells are functions between the underlying types and 2-cells are homotopies between functions.

2 Quotient and Symmetric Containers

In this section we recall the notions of quotient and symmetric containers, as well as their morphisms.

2.1 Quotient Containers

Quotient containers were introduced by Abbott et al. [3] as a way to add symmetries to containers in an extensional type theory with quotient types. Here we state their definition in HoTT.

▶ **Definition 1.** *A* quotient container $(S \triangleright P/G)$ *consists of a set of* shapes S*, a family of* positions $P : S \to \mathsf{hSet}$ *and* symmetry groups $\iota_s : G_s \leq \mathfrak{S}(P_s)$ *for each* $s : S$*.*

Each group element $g : G_s$ defines a path $\iota_s(g) : P_s = P_s$. By $\mathsf{transport}$, this induces a map $P_s \to P_s$; in the remainder, we will identify g and this map.

▶ **Example 2.** The quotient container of *unordered n-tuples* $\mathsf{U}_n := (1 \triangleright \mathsf{Fin}(n)/\mathfrak{S}(n))$ has a single shape, and over it positions $\mathsf{Fin}(n)$. On these n positions, the full group of permutations $\mathfrak{S}(\mathsf{Fin}(n))$ acts as its symmetry group. We call U_1 the *identity container*; it has a single shape, on which the trivial group acts.

Like an ordinary container, a quotient container defines an endofunctor on the category of sets, called its *extension*. Whereas for ordinary containers this is a polynomial functor, for quotient containers it is a sum of quotients of representables:

▶ **Definition 3** (extension of quotient containers)**.** *The extension of* $(S \triangleright P/G)$ *is the map* $[\![S \triangleright P/G]\!]_/ : \mathsf{hSet} \to \mathsf{hSet}$ *given by*

$$[\![S \triangleright P/G]\!]_/(X) := \sum_{s:S} \frac{P_s \to X}{\sim_s}, \qquad\qquad v \sim_s w := \exists_{g:G_s}\, v = w \circ g \qquad ⌟$$

When positions are finite sets, the extension is an *analytic functor* in the sense of Joyal [15].

▶ **Example 4** (unordered n-tuples)**.** The extension of U_n is the type of unordered n-tuples:

$$[\![\mathsf{U}_n]\!]_/(X) := \sum_{\_:1} (\mathsf{Fin}(n) \to X)/\!\sim\, = X^n/\!\sim_{\mathfrak{S}(n)}$$

where $x \sim_{\mathfrak{S}(n)} y$ if and only if $x_i = y_{\sigma(i)}$ for some permutation $\sigma : \mathfrak{S}(n)$. When $n = 1$, we obtain the identity function $[\![\mathsf{U}_1]\!]_/ X = X$.

▶ **Definition 5.** *A* premorphism *of quotient containers,* $(u \triangleright f) : (S \triangleright P/G) \rightharpoonup (T \triangleright Q/H)$ *consists of a map of shapes* $u : S \to T$*, a map of positions* $f : \prod_{s:S} Q_{us} \to P_s$*, and a proof that* f preserves symmetries*,* $\prod_{s:S} \prod_{g:G_s} \exists_{h:H_{us}}\, g \circ f_s = f_s \circ h$*.*

Morphisms of quotient containers are defined up to permutation of positions, i.e. as equivalence classes of premorphisms.

▶ **Definition 6.** *The type of* morphisms $F \to G$ *of quotient containers is the set quotient of the type of premorphisms $F \rightharpoonup G$ by the relation (defined by path induction)*

$$(u \triangleright f) \approx (u' \triangleright f') := \sum_{p:u=u'} f \approx'_p f', \qquad f \approx'_{\mathsf{refl}_u} f' := \prod_{s:S} \exists_{h:H_{us}} f_s = f'_s \circ h \qquad \lrcorner$$

Whenever $u \doteq u'$, this relation posits the mere existence of a triangle filler

$$(u \triangleright f) \approx (u \triangleright f') \quad \simeq \quad \begin{tikzcd} Q_{us} \arrow[rr, dashed, "\exists(h:H_{us})"] \arrow[dr, "f_s"'] & & Q_{us} \arrow[dl, "f'_s"] \\ & P_s & \end{tikzcd}$$

▶ **Definition 7.** *Quotient containers and their morphisms form a category* QuotCont*.*

Extension of quotient containers is a functor $[\![-]\!]_/ : \mathsf{QuotCont} \to \mathsf{Endo}(\mathsf{hSet})$, which is full and faithful. Each premorphism $(u \triangleright f) : F \rightharpoonup G$ defines a natural transformation $[\![u \triangleright f]\!]_/ : [\![F]\!]_/ \Rightarrow [\![G]\!]_/$, with component at $X : \mathsf{hSet}$ a map

$$[\![u \triangleright f]\!]^X_/ : \sum_{s:S}(P_s \to X/{\sim_s}) \to \sum_{t:T}(Q_t \to X/{\sim_t})$$

defined by induction on set quotients as $[\![u \triangleright f]\!]^X_/(s, [\ell]) := (us, [\ell \circ f])$. This is well-defined on morphisms of quotient containers: If $(u \triangleright f) \approx (u' \triangleright f')$ then $[\![u \triangleright f]\!]^X_/ = [\![u' \triangleright f']\!]^X_/$.

2.2 Symmetric Containers

Symmetric containers were introduced by Gylterud [10] as a way of defining polynomial functors between categorical groupoids. In this section we reformulate their definition in the language of HoTT using homotopy groupoids instead.

▶ **Definition 8.** *A* symmetric container $(S \triangleleft P)$ *consists of shapes $S : \mathsf{hGpd}$ and a family of positions $P : S \to \mathsf{hSet}$.*

▶ **Definition 9.** *A morphism of symmetric containers $(u \triangleleft f) : (S \triangleleft P) \to (T \triangleleft Q)$ consists of a map of shapes $u : S \to T$ and a family $f : \prod_{s:S} Q_{us} \to P_s$ of maps of positions.*

In a (homotopy) type-theoretic setting, the types of morphisms $\mathsf{C}(x, y)$ of a category C are understood to be h-sets. Morphisms of symmetric containers, however, form h-groupoids.[1]

▶ **Definition 10.** *The 2-category* SymmCont *has as objects symmetric containers, as 1-cells morphisms of symmetric containers, and as 2-cells identifications of such morphisms.*

▶ **Definition 11.** *The extension of a symmetric container $(S \triangleleft P)$ is a function of h-groupoids, $[\![S \triangleleft P]\!] : \mathsf{hGpd} \to \mathsf{hGpd}$, defined as $[\![S \triangleleft P]\!](X) := \sum_{s:S} P_s \to X$.*

Extension of symmetric containers defines a 2-functor $[\![-]\!] : \mathsf{Symm} \to \mathsf{Endo}(\mathsf{hGpd})$. For any morphism $(u \triangleleft f) : (S \triangleleft P) \to (T \triangleleft Q)$, there is a pseudonatural transformation $[\![u \triangleleft f]\!] : [\![S \triangleleft P]\!] \Rightarrow [\![T \triangleleft Q]\!]$ with components given by precomposition

$$[\![u \triangleleft f]\!]_X : \sum_{s:S}(P_s \to X) \to \sum_{t:T}(Q_t \to X) \qquad [\![u \triangleleft f]\!]_X(s, v) := (us, Q_{us} \xrightarrow{f_s} P_s \xrightarrow{v} X)$$

[1] Observe that $(S \triangleleft 0) \to (T \triangleleft 0) \simeq (S \to T)$, which is an h-groupoid since T is.

This 2-functor is locally an equivalence of 2-categories [10, Theorem 2.2.1], thus embeds symmetric containers into endofunctors of groupoids.

One advantage of internalizing symmetric containers as h-groupoids is that we are free to define groupoids of shapes as higher inductive types, encoding the desired symmetries directly in their constructors. For example, cyclic lists can be described using the symmetries of the circle, S^1:

▶ **Example 12.** The symmetric container of *cyclic lists*, Cyc, is defined as follows:

- Shapes are pairs $\mathbb{N} \times S^1$. The first component contains the length of the list. The second has a single point, $\mathsf{base} : S^1$, but its loops $\mathsf{base} = \mathsf{base}$ are going to induce cyclic shifts on positions.
- Positions $\mathsf{Sh} : \mathbb{N} \times S^1 \to \mathsf{hSet}$ are defined by induction on the circle S^1. Over the point, we have n distinct positions, $\mathsf{Sh}(n, \mathsf{base}) := \mathsf{Fin}(n)$. On the non-trivial path, Sh identifies positions by a cyclic shift, $\mathsf{Sh}(\mathsf{refl}_n, \mathsf{loop}) := \mathsf{shift}$. Here, $\mathsf{shift} : \mathsf{Fin}(n) = \mathsf{Fin}(n)$ is the path obtained by univalence from the *successor* equivalence

$$\begin{aligned} \mathsf{suc} &: \mathsf{Fin}(n) \simeq \mathsf{Fin}(n) \\ \mathsf{suc}(k) &:= (k+1) \mod n \end{aligned}$$

Since S^1 is freely generated by a single non-trivial path, Sh identifies positions by arbitrary cyclic shifts. Let $v := (x, y, z)$, $w := (y, z, x)$ terms of type $\mathsf{Fin}(3) \to X$. We are going to exhibit a path $\big((3, \mathsf{base}), v\big) = \big((3, \mathsf{base}), w\big)$ in $[\![\mathsf{Cyc}]\!](X)$. Since $[\![\mathsf{Cyc}]\!](X)$ is an iterated Σ-type, such a path is given by a triple of paths $p : 3 = 3$, $q : \mathsf{base} = \mathsf{base}$, and $r : v =_q w$. Set $p := \mathsf{refl}$ and $q := \mathsf{loop} \cdot \mathsf{loop}$. The path r is dependent over q, and it suffices to give a path $v = w \circ \mathsf{shift} \circ \mathsf{shift}$. But we see that the right side computes to (x, y, z), which is v.

2.3 Lifting Quotient Containers

The data of both quotient and symmetric containers define semantics for datatypes with symmetries. As we will see, it is possible to see any quotient container as a symmetric one. However, this analogy does not extend to (polymorphic) functions between such types, since it is generally not possible to lift a morphism of quotient containers to one of symmetric containers, as the former truncate evidence on how symmetries are preserved.

In order to create a symmetric container from a quotient container, we have to come up with a *groupoid* of shapes that encodes the symmetries present in the quotient container. We borrow an idea from algebraic topology: any group G gives rise to a unique pointed, connected groupoid $(\mathbf{B}G, \bullet)$ such that $\Omega(\mathbf{B}G, \bullet) \simeq G$, called its *delooping*. This type is an instance of an Eilenberg–MacLane space, i.e. a type with a single non-trivial homotopy group. Eilenberg–MacLane spaces have been studied in HoTT [16], and our presentation of $\mathbf{B}G$ coincides with $K(G, 1)$ there. We define $\mathbf{B}G$ as a *higher inductive type* with constructors

$$\frac{}{\bullet : \mathbf{B}G} \qquad \frac{g : G}{\mathsf{loop}\, g : \bullet = \bullet} \qquad \frac{g, h : G}{\mathsf{loop\text{-}comp}(g, h) : \mathsf{loop}\, g \cdot \mathsf{loop}\, h = \mathsf{loop}\, gh}$$

plus one constructor asserting that $\mathbf{B}G$ is an h-groupoid. Its *recursion principle* states that, to define a map $\mathbf{B}G \to X$ for some groupoid X, it suffices to give a point $x_0 : X$ and a group homomorphism $\varphi : G \xrightarrow{\cdot} \Omega\,(X, x_0)$: a map $f : \mathbf{B}G \to X$ is then determined by $f(\bullet) := x_0$ and $f(\mathsf{loop}\, g) := \varphi(g)$. The recursor characterizes functions out of $\mathbf{B}G$, in the following sense:

▶ **Proposition 13.** *The recursor is an equivalence* $(\sum_{x_0 : X} G \xrightarrow{\cdot} \Omega\,(X, x_0)) \simeq (\mathbf{B}G \to X)$*, for* X *a groupoid. When* X *is a set, we have* $(\sum_{x_0 : X} G \to x_0 = x_0) \simeq (\mathbf{B}G \to X)$*.*

The *dependent eliminator* lets us define sections $\prod_{x:\mathbf{B}G} B(x)$ of families $B : \mathbf{B}G \to \mathsf{hGpd}$ by providing a point $b_0 : B(\bullet)$, dependent paths $\varphi : \prod_{g:G}(b_0 =_{B(\mathsf{loop}\, g)} b_0)$, and a coherence condition for composition of dependent paths: for all $g, h : G$, there needs to be a path from $\varphi(g) \cdot \varphi(h)$ to $\varphi(gh)$, dependent over $\mathsf{loop\text{-}comp}(g, h)$.

Note that $\mathsf{loop\text{-}comp} : G \to \Omega(\mathbf{B}G, \bullet)$ preserves the product of G, hence is a morphism of groups. This lets us derive other expected coherences, such as $\mathsf{loop}\, 1_G = \mathsf{refl}$ and $\mathsf{loop}\,(g^{-1}) = (\mathsf{loop}\, g)^{-1}$.

Delooping acts on group homomorphisms:

▶ **Definition 14.** *Any group homomorphism $\varphi : G \to H$ induces a map of groupoids $\mathbf{B}\varphi : \mathbf{B}G \to \mathbf{B}H$, defined by induction:*

$$\mathbf{B}\varphi(\bullet) := \bullet, \qquad\qquad \mathbf{B}\varphi(\mathsf{loop}\, g) := \mathsf{loop}\, \varphi(g)$$

A G-action is a particular homomorphism, so the above defines a type family $\mathbf{B}G \to \mathsf{hSet}$. Let us spell this out:

▶ **Definition 15** (associated bundle)**.** *Let G act on a set X via $\sigma : G \dashrightarrow \mathfrak{S}(X)$. Its* associated bundle *$\bar{\mathbf{B}}\sigma : \mathbf{B}G \to \mathsf{hSet}$ is defined by recursion on $\mathbf{B}G$ as*

$$\bar{\mathbf{B}}\sigma(\bullet) := X, \qquad\qquad \bar{\mathbf{B}}\sigma(\mathsf{loop}\, g) := \sigma(g),$$

Note that for each $g : G$, $\sigma(g)$ is a path $X = X$.

In the context of quotient containers, we are dealing with *faithful* group actions, that is actions of G on X such that $\sigma : G \to \mathfrak{S}(X)$ is an embedding. In this case, the associated bundle is an embedding on its path spaces, i.e. a set-truncated function [20, 7.6.1]:

▶ **Proposition 16.** *If $\sigma : G \hookrightarrow \mathfrak{S}(X)$ acts* faithfully*, the fibers of $\bar{\mathbf{B}}\sigma : \mathbf{B}G \to \mathsf{hSet}$ are sets.*

Proof. By [20, Lemma 7.6.2], $\bar{\mathbf{B}}\sigma$ has set-valued fibers iff $\mathsf{cong}\,\bar{\mathbf{B}}\sigma : x = y \to \bar{\mathbf{B}}\sigma(x) = \bar{\mathbf{B}}\sigma(y)$ is an embedding for all $x, y : \mathbf{B}G$. This is a proposition, therefore it suffices to show this at $x \doteq y \doteq \bullet$. There, we have $\mathsf{cong}(\bar{\mathbf{B}}\sigma) \circ \mathsf{loop} \doteq \sigma$. But $\mathsf{loop} : G \to \bullet = \bullet$ is an equivalence and σ an embedding, hence $\mathsf{cong}\,\bar{\mathbf{B}}\sigma$ is an embedding as well. ◀

For any quotient container we define a groupoid that is the collection of its delooped symmetry groups:

▶ **Definition 17.** *The* delooping *of a quotient container $(S \triangleright P/G)$ is the symmetric container $\mathbf{B}(S \triangleright P/G) := (\mathbf{S} \triangleleft \mathbf{P})$ consisting of*

- *shapes $\mathbf{S} := \sum_{s:S} \mathbf{B}G_s$, and*
- *positions $\mathbf{P} : \sum_{s:S} \mathbf{B}G_s \to \mathsf{hSet}$, $\mathbf{P}(s, -) := \bar{\mathbf{B}}(\iota_s)$,*

where ι_s is the inclusion of symmetry groups $G_s \hookrightarrow \mathfrak{S}(X)$.

We think of the shapes $\mathbf{S}$ as consisting of the points of S, with loops given by elements in G_s freely added. Indeed, if we compute its connected components, we see that

$$\pi_0(\mathbf{S}) \simeq \|\textstyle\sum_{s:S} \mathbf{B}G_s\|_0 \simeq \sum_{s:S} \|\mathbf{B}G_s\|_0 \simeq S$$

where the last step follows from connectivity of $\mathbf{B}G_s$. Similarly, we compute its first fundamental group: S is a set and $s = s$ is contractible, thus

$$\pi_1(\mathbf{S}, (s, x)) \simeq \textstyle\sum_{p:s=s}(x =_p x) \simeq (x = x) \simeq \pi_1(\mathbf{B}G_s, x) \simeq G_s$$

That each G_s acts faithfully is reflected in the action of the groupoid $\mathbf{S}$, in the sense that the bundle $\mathbf{P}$ embeds paths of $\mathbf{S}$:

▶ **Proposition 18.** *The family* $\mathbf{P} : \mathbf{S} \to \mathsf{hSet}$ *is a set-truncated function.*

Proof. For each $X : \mathsf{hSet}$, we have $\mathsf{fiber}_{\mathbf{P}}\, X \simeq \sum_{s:S} \mathsf{fiber}_{\bar{\mathbf{B}}\iota_s}\, X$, which is a set: S is a set, and since we assume ι_s to be faithful, so are the fibers of $\bar{\mathbf{B}}\iota_s$ by Proposition 16. ◀

This way of obtaining a symmetric container is in some sense conservative: when comparing the associated extensions (and set-truncating that of the symmetric container), we see that they are the same function of sets:

▶ **Theorem 19.** *For a quotient container* Q *and* $X : \mathsf{hSet}$*, there is an equivalence of sets*

$$\| \llbracket \mathbf{B}Q \rrbracket(X) \|_0 \simeq \llbracket Q \rrbracket_{/}(X)$$

Proof. Let us unfold the definitions of $\llbracket - \rrbracket$ and $\mathbf{B}$,

$$\| \llbracket \mathbf{B}Q \rrbracket(X) \|_0 \simeq \left\| \textstyle\sum_{s:S} \sum_{x:\mathbf{B}G_s} \bar{\mathbf{B}}\iota_s(x) \to X \right\|_0 \tag{1}$$

and, as S is a set, move the truncation under the sum

$$\simeq \textstyle\sum_{s:S} \left\| \sum_{x:\mathbf{B}G_s} \bar{\mathbf{B}}\iota_s(x) \to X \right\|_0 \tag{2}$$

Notice that G_s acts on the function type $P_s \to X$ via precomposition, and that its associated bundle is $(\bar{\mathbf{B}}\iota_s(-) \to X) : \mathbf{B}G_s \to \mathsf{hSet}$. By Lemma 20 below, the connected components of this bundle correspond to orbits of the action,

$$\simeq \sum_{s:S} (P_s \to X)/G_s \tag{3}$$

which is exactly how extension of a quotient container is defined:

$$\simeq \sum_{s:S} (P_s \to X)/{\sim_s} \simeq \llbracket Q \rrbracket_{/}(X)$$ ◀

▶ **Lemma 20.** *Let* σ *a* G*-action on* X*. The connected components of the total space of its associated bundle and* σ*-orbits are in bijection, that is* $\| \sum_{x:\mathbf{B}G} \bar{\mathbf{B}}\sigma(x) \|_0 \simeq X/G$*.*

Proof. Let us define the left-to-right direction. Since the codomain is a set, it suffices to give $f : \prod_{x:\mathbf{B}G} \bar{\mathbf{B}}\sigma(x) \to X/G$ by induction on $\mathbf{B}G$. Let $f(\bullet) := [-] : X \to X/G$ the surjection onto the quotient. It remains to show that this is well-defined on loops, which reduces to $\prod_{g:G} \prod_{x:X} [x] = [\sigma_g(x)]$. This holds since $x \sim \sigma_g(x)$ by definition of the orbit relation. The inverse is defined by recursion on the set quotient X/G and maps $x : X$ to $(\bullet, x) : \sum_{x:\mathbf{B}G} \bar{\mathbf{B}}\sigma(x)$. From $p : x = \sigma_g(y)$, one constructs a path $(\bullet, x) = (\bullet, y)$: the first component is given by $\mathsf{loop}\, g$, the second as dependent path from p. ◀

Theorem 19 states that, as functions between types, the diagram

$$\begin{tikzcd}
\mathsf{SymmCont} \arrow[r, "\llbracket - \rrbracket"] & (\mathsf{hGpd} \to \mathsf{hGpd}) \arrow[d, "\lambda F. \| F(-) \|_0"] \\
\mathsf{QuotCont} \arrow[u, "\mathbf{B}"] \arrow[r, "\llbracket - \rrbracket_{/}"'] & (\mathsf{hSet} \to \mathsf{hSet})
\end{tikzcd}$$

commutes. We are interested to see whether this generalizes to a natural isomorphism of functors. To do so, we would have to suitably extend $\mathbf{B}$ to a functor. Unfortunately, it is not clear how to define its action on morphisms of quotient containers. Given a premorphism $(u \triangleright f) : (S \triangleright P/G) \rightharpoonup (T \triangleright Q/H)$, we would have to provide a morphism of shapes

$$\begin{aligned} &\textstyle\sum_{s:S} \mathbf{B}G_s \to \sum_{t:T} \mathbf{B}H_t \\ &\qquad (s, x) \mapsto (us, ?) \end{aligned}$$

To define $? : \mathbf{B}G_s \to \mathbf{B}H_{us}$, it would suffice to provide a morphism of groups $G_s \dot{\to} H_{us}$. However, we are not given this information: we know that f preserves symmetries, but this only tells us that, for each $g : G_s$, there is *merely* some $h : H_{us}$. Even if we were given an explicit function $G_s \to H_{us}$, it would not have to be a group homomorphism. In fact, it is easy to construct counterexamples:

▶ **Example 21.** Consider $(\mathsf{id} \triangleright !) : \mathsf{U}_1 \rightharpoonup \mathsf{U}_2$. The terminal map $! : 2 \to 1$ trivially preserves symmetries: the diagram

$$\begin{tikzcd} 2 \arrow[r, "!"] \arrow[d, "\varphi_g"'] & 1 \arrow[d, "g"] \\ 2 \arrow[r, "!"'] & 1 \end{tikzcd}$$

commutes for any choice of φ_g, in particular for $\varphi : \mathfrak{S}(1) \to \mathfrak{S}(2), \varphi_{\_} := \mathsf{swap}$, which is *not* a group homomorphism.

Since morphisms of quotient containers are equivalence classes, it might be possible to find another premorphism in the same class for which this assignment is a morphism of groups. In fact, in the above example one could pick $\varphi_g := \mathsf{id}$, which clearly is. Doing so however for *arbitrary* symmetry groups seems constructively impossible, without invoking some form of choice principle.

Instead, we will be looking to enhance the definition of quotient containers to include the necessary information, and investigate their relation to symmetric containers more closely.

3 Action Containers

In this section we define *action containers* and assemble them into a 1-category. Morphisms in this category are akin to premorphisms of quotient containers. In particular, they are not quotiented by a relation on positions. Later, in Section 4, we enrich this category to obtain a (2,1)-category whose 2-cells capture this relation.

Different from quotient containers, the symmetries of action containers are not limited to subgroups of permutations of positions. Instead, an action container has, for each shape, a chosen group *acting* on the set of positions. This lets us flexibly introduce symmetries, e.g. by letting the integers under addition act on a finite set, instead of having to identify the image of this action, see the forthcoming Example 23.

The category of action containers admits a number of limits and colimits, and we will derive the usual container algebra of products and coproducts from a presentation of this category as a category of families in Section 3.1.

▶ **Definition 22.** *An* action container $(S \triangleright P \triangleleft^{\sigma} G)$ *consists of a set of* shapes S*, a family of* positions $P : S \to \mathsf{hSet}$ *and* group actions $\sigma_s : G_s \to \mathfrak{S}(P_s)$ *for each* $s : S$*.*

In the following, the word "container" refers to action containers; other kinds of containers are qualified explicitly. In Example 12, we defined cyclic lists as a symmetric container in which loops of the circle act by cyclic shifts. Since $\Omega(S^1) = \mathbb{Z}$, we are inspired to define a container of cyclic lists by means of a $\mathbb{Z}$-action:

▶ **Example 23** (cyclic lists as an action container)**.** The container $\mathsf{Cyc} := (\mathbb{N} \triangleright \mathsf{Fin} \triangleleft^{\sigma} \mathbb{Z})$ has $\mathbb{Z}$ acting on $\mathsf{Fin}(n)$ as follows: for each n, let $\sigma_n : \mathbb{Z} \to \mathfrak{S}(n)$, $\sigma_n(k) := \lambda \ell.\, (\ell + k) \mod n$. Note that this action is *not* faithful: the kernel of σ_n consists of the integers $n\mathbb{Z} = \{\, nk \mid k \in \mathbb{Z} \,\}$.

In general, it is easy to define $\mathbb{Z}$-actions: $\mathbb{Z}$ is the free group on one generator, thus it suffices to define the action of $1 : \mathbb{Z}$. In the example of cyclic lists, it suffices to define the cyclic shift by one position, $\sigma_n(1) := \lambda \ell.\, (\ell + 1) \mod n$. This is impossible for quotient containers with finitely many positions: $\mathbb{Z}$ is simply never a subgroup of finite symmetry groups.

Unlike premorphisms of quotient containers, morphism of action containers are required to preserve the full structure of containers, including their symmetries:

▶ **Definition 24.** *A* morphism *of action containers* $(u \triangleright f \triangleleft \varphi) : (S \triangleright P \triangleleft^{\sigma} G) \to (T \triangleright Q \triangleleft^{\tau} H)$ *consists of a map of shapes* $u : S \to T$*, a map of positions* $f : \prod_{s:S} Q_{us} \to P_s$*, a family of group homomorphisms* $\varphi : \prod_{s:S} G_s \dashrightarrow H_{us}$*, and a proof that* f *is* equivariant*: for all* $s : S$ *and* $g : G_s$ *a commutative square*

$$\begin{tikzcd}
Q_{us} \arrow[r, "f_s"] \arrow[d, "\tau_{us}(\varphi_s(g))"'] & P_s \arrow[d, "\sigma_s(g)"] \\
Q_{us} \arrow[r, "f_s"'] & P_s
\end{tikzcd}$$

Calling f equivariant is justified: each f_s is a morphism between G_s-sets (P_s, σ_s) and $(Q_{us}, \tau_{us} \circ \varphi_s)$ [18, Definition 1.2]. Theorem 27 will explain how this notion of morphism arises naturally from a category of group actions and equivariant maps between them.

▶ **Definition 25.** *Action containers and their morphisms form a category* $\mathsf{ActCont}$.

3.1 Algebra of Action Containers

Like other categories of containers, action containers are closed under a number of constructions. In particular, $\mathsf{ActCont}$ has all products and coproducts. To show this, we could define each of them by hand. In that process, we would have to carefully track the variance of parts of a container. For example, the binary product of containers is a product of shapes and symmetry groups, but a (pointwise) coproduct of families of positions.

Instead, we opt to present $\mathsf{ActCont}$ as a category of *families of group actions*, from which (co)limits are easy to read off. First, we define a category of group actions. It is a version of the category of G-sets (for a fixed group G), in which equivariant maps are permitted to go between sets with actions of *different groups*.

▶ **Definition 26.** *We denote by* Action *the category of group actions and equivariant maps. It is obtained as the total category of the following category displayed over* $\mathsf{Group} \times \mathsf{hSet}^{\mathsf{op}}$*:*

- *Given objects* $G : \mathsf{Group}_0$ *and* $X : \mathsf{hSet}_0^{\mathsf{op}}$*, displayed objects are* G*-actions on* X*, i.e. group homomorphisms* $\mathsf{Action}_0(G, X) := (G \dashrightarrow \mathfrak{S}(X))$*.*
- *Displayed morphisms between actions* $\sigma : \mathsf{Action}_0(G, X)$ *and* $\tau : \mathsf{Action}_0(H, Y)$ *over* $(\varphi : G \dashrightarrow H, f : X \leftarrow Y)$ *are proofs of the proposition "*f *is equivariant over* φ*": Let* $\mathsf{Action}_1((\varphi, f); \sigma, \tau) := \mathsf{isEquivariant}_{\varphi,f}(\sigma, \tau) := \prod_{g:G} \sigma(g) \circ f = f \circ \tau(\varphi g)$*.*

Diagrammatically, equivariant maps compose by horizontal pasting of proofs of equivariance:

$$\begin{tikzcd} X \arrow[d, "\sigma(g)"'] & Y \arrow[l, "f"'] \arrow[d, "\tau(\varphi g)"] \\ X & Y \arrow[l, "f"] \end{tikzcd} \; ; \; \begin{tikzcd} Y \arrow[d, "\tau(h)"'] & Z \arrow[l, "e"'] \arrow[d, "\upsilon(\psi h)"] \\ Y & Z \arrow[l, "e"] \end{tikzcd} \; := \; \begin{tikzcd} X \arrow[d, "\sigma(g)"'] & Z \arrow[l, "fe"'] \arrow[d, "\upsilon(\psi\varphi g)"] \\ X & Z \arrow[l, "fe"] \end{tikzcd}$$

Observe how over each shape, the data of a container $(S \triangleright P \triangleleft^{\sigma} G)$ is exactly that of a group action: for any $s : S$, G_s acts on P_s via σ_s. Thus, on objects, the category of action containers consists of "families" of group actions. Let us ensure that this analogy extends to morphisms of this category.

Recall that for any category $\mathcal{C}$, its free coproduct completion is the category of families $\mathrm{Fam}(\mathcal{C})$ [4, §2]. Its objects are families $\sum_{I:\mathsf{hSet}} I \to \mathcal{C}_0$, morphisms are families of maps between them.

▶ **Theorem 27.** *The category of action containers is equivalent to families of group actions. In particular, the functor* $\mathsf{F} : \mathsf{ActCont} \to \mathrm{Fam}(\mathsf{Action})$ *with action on objects given by* $\mathsf{F}(S \triangleright P \triangleleft^{\sigma} G) := (S, \lambda s.\, (G_s, P_s, \sigma_s))$ *is an equivalence of categories.*

▶ Remark (univalence of ActCont). The above implies that ActCont is a univalent category: Fam preserves univalence, and Action is a univalent displayed category by [7, Proposition 7.3].

From this presentation of ActCont, we read off closure under sums and products:

▶ **Proposition 28.** Action *has K-indexed products for all sets K. In particular, the trivial group acting on the singleton set is an initial object.*

▶ **Corollary 29.** *Action containers are closed under products and coproducts.*

Proof. $\mathrm{Fam}(\mathcal{C})$ is the free coproduct completion of any category $\mathcal{C}$, thus ActCont is closed under coproducts. Similarly, Action is closed under products (Proposition 28), and Theorem 2.11 of [4] implies that families over it are as well. ◀

Like ordinary containers [2, Proposition 3.9], constant action containers are exponentiable:

▶ **Proposition 30** (constant containers are exponentiable)**.** *The* constant container *of a set K is* $\mathbf{k}K := (K \triangleright 0 \triangleleft^{\mathrm{id}} 1)$. *Given a container $F = (S \triangleright P \triangleleft^{\sigma} G)$, the exponential container* $F^K := (S^* \triangleright P^* \triangleleft^{\sigma^*} G^*)$ *is defined to have*

- *shapes* $S^* := K \to S$,
- *positions* $P^*_f := \sum_{k:K} P_{fk}$,
- *symmetries* $G^*_f := \prod_{k:K} G_{fk}$, *and*
- *actions* $\sigma^*_f : G^*_f \to \mathfrak{S}(P^*_f)$ *given by action of σ on the second component of P^*_f: for* $g : \prod_k G_{fk}$, *let* $\sigma^*_f(g) := \lambda(k,p).\,(k, \sigma_{fk}(g))$

Let $f : K \to S$ and $k : K$. The evaluation morphism $\mathrm{ev} : F^K \times \mathbf{k}K \to F$ *is given by function application $fk : S$ on shapes, pairing* $P_{fk} \to 0 + \sum_k P_{fk}$ *on positions, and the projection homomorphisms* $1 \times \prod_{k'} G_{fk'} \dashrightarrow G_{fk}$ *on symmetries.*

We believe that the above is an instance of constant exponentials in families: Let C have K-fold products for any set K; in particular an initial object 1_{C} and binary products. It should be possible to show that the constant family $(K, \lambda k.\, 1_{\mathsf{C}})$ is exponentiable.

▶ Remark (composition of action containers). When seen as endofunctors, composite data types are constructed as compositions of functors. For ordinary containers [2, 3.6] and symmetric containers [10, 2.2.5], this construction is reflected along $[\![-]\!]$: Given containers F and G, there is a container $F[G]$ such that $[\![F[G]]\!] \cong [\![F]\!] \circ [\![G]\!]$, upgrading $[\![-]\!]$ to a strong monoidal functor. In a set-theoretic setting, Hasegawa [11, Corollary 1.18] gives a construction of composition for analytic functors in terms of wreath products of symmetry groups. Analytic functors are presented by quotient containers with *finitary* positions, and such a restriction of positions to a subuniverse of sets seems necessary when porting the result to action containers: positions and symmetries of the composite can otherwise not be defined coherently by induction on set quotients. This indicates to us that the definition of a composition-monoidal structure of action containers is not entirely straightforward, and we postpone its study for now.

4 The 2-Category of Action Containers

In Section 2.3 we observed that quotient containers lift to symmetric containers, but that the same does not apply to their morphisms. We defined the category of action containers to include the missing data, and are now ready to define an appropriate lifting:

▶ **Proposition 31.** *The* delooping *of a container* $(S \triangleright P \triangleleft^{\sigma} G)$ *is the symmetric container* $\mathbf{B}(S \triangleright P \triangleleft^{\sigma} G) := (\sum_{s:S} \mathbf{B}G_s \triangleleft \bar{\mathbf{B}}\sigma_s)$*. Each morphism* $(u \triangleright f \triangleleft \varphi) : (S \triangleright P \triangleleft^{\sigma} G) \to (T \triangleright Q \triangleleft^{\tau} H)$ *defines a morphism between deloopings,* $(\bar{\varphi} \triangleleft \bar{f}) : \mathbf{B}(S \triangleright P \triangleleft^{\sigma} G) \to \mathbf{B}(T \triangleright Q \triangleleft^{\tau} H)$*.*

Proof. The family φ yields a map of shapes of type $\bar{\varphi} : \sum_{s:S} \mathbf{B}G_s \to \sum_{t:T} \mathbf{B}H_t$, defined as $\bar{\varphi}(s, x) := (us, \mathbf{B}\varphi_s(x))$. The map on positions has (uncurried) type

$$\bar{f} : \prod_{s:S} \prod_{x:\mathbf{B}G_s} \bar{\mathbf{B}}\tau_{us}(\mathbf{B}\varphi(x)) \to \bar{\mathbf{B}}\sigma_s(x)$$

and is defined by induction on $x : \mathbf{B}G_s$. On the point, let $\bar{f}(s, \bullet) := f_s : Q_{us} \to P_s$. It remains to show that $\bar{f}$ is well-defined on loops in $\mathbf{B}G_s$. For all g, we have to provide a dependent path $f_s =_{F(\mathsf{loop}\, g)} f_s$ where $F(x) = \bar{\mathbf{B}}\tau_{us}(\mathbf{B}\varphi(x)) \to \bar{\mathbf{B}}\sigma_s(x)$. By [20, Lemma 2.9.6], this is equivalent to giving paths $\prod_{q:Q_{us}} \sigma_s(f_s(q)) = f_s((\tau_{us}\varphi_s\, g)\, q)$, which we obtain from the proof that f_s is equivariant. ◀

As noted in Section 2.2, symmetric containers do not form an ordinary category, but a 2-category. Thus, in order to show that the above construction is functorial, we must first enrich action containers by a type of 2-cells, defining a 2-category. We do so by pulling back 2-cells of symmetric containers, and will see that this corresponds to the quotiented sets of quotient container morphisms.

We split the construction of the 2-functor taking action containers into symmetric containers into smaller steps. As in Section 3, we first seek to understand the problem for a single action, before considering entire families of such. We observe that symmetric containers, from a homotopical viewpoint, are *set-bundles over groupoids*: both consist of some *base* B : hGpd together with a family of *fibers* $F : B \to$ hSet. Previously, we have seen that each action defines such a bundle, namely its *associated bundle* (Definition 15). We define 2-categories of actions (Action, Definition 37) and set bundles (Definition 38), and show that taking the associated bundle is a weak equivalence of their local categories (Theorem 42). This means that maps of actions are in 1-to-1 correspondence with functions on their associated bundles. By changing our point of view, and seeing actions as single-shape containers and bundles as symmetric containers, this fully classifies morphisms of (single-shape) action containers in terms of morphisms of symmetric containers.

To understand the case of many-shape containers, we give an analogue of the Fam-construction in 2-categories, and define the 2-category of action containers as $\mathsf{ActCont} :=$ Fam(Action). The objects and 1-cells of this category are exactly as in Definition 26. We unfold the type of 2-cells induced by this construction (Proposition 47) and show that it is closely related to the quotient on premorphisms of quotient containers (Definition 6). We observe that set-bundles are, in a suitable sense, closed under Σ-types, and we lift the functor $\bar{\mathbf{B}} : \mathsf{Action} \to \mathsf{SetBundle}$ to

$$\mathsf{ActCont} \xrightarrow{=} \mathrm{Fam}(\mathsf{Action}) \xrightarrow{\mathrm{Fam}(\bar{\mathbf{B}})} \mathrm{Fam}(\mathsf{SetBundle}) \xrightarrow{\Sigma} \mathsf{SetBundle} \xrightarrow{=} \mathsf{SymmCont}$$

finally establishing the connection between action containers and symmetric containers.

4.1 A 2-Category of Groups

We think of the type of h-groupoids, hGpd, as an internal notion of categorical groupoids: The 2-category hGpd has as objects h-groupoids, as morphisms functions of such, and as 2-cells homotopies between them. Our goal is to extend the delooping to a 2-functor taking groups into hGpd in a way that characterizes 2-cells in hGpd. We thus equip the 1-category of groups with the structure of a (2,1)-category [12]:

▶ **Definition 32.** *The category* Group *of groups and group homomorphisms forms a (2,1)-category if equipped with the following 2-cells: Let* $\varphi, \psi : \mathsf{Group}_1(G, H)$*. We say that* $r : H$ *is a* conjugator *of* φ *and* ψ *if*

$$\mathsf{isConjugator}_{\varphi,\psi}(r) := \prod_{g:G} \varphi(g) \cdot r = r \cdot \psi(g)$$

The 2-cells $\mathsf{Group}_2(\varphi, \psi) := \sum_{r:H} \mathsf{isConjugator}_{\varphi,\psi}(r)$ *compose vertically by multiplication in* H*. The horizontal composites of* $r : \mathsf{Group}_2(\varphi, \psi)$ *and* $s : \mathsf{Group}_2(\varphi', \psi')$ *is* $s \cdot \psi'(r)$*.*

Note that Group is not locally univalent: the identity type of group homomorphisms, $\varphi = \psi$, is a proposition, but the type of conjugators $\mathsf{Group}_2(\varphi, \psi)$ is a set.

▶ **Lemma 33.** *Delooping extends to a 2-functor* $\mathbf{B} : \mathsf{Group} \to \mathsf{hGpd}$*.*

Proof. A 1-cell $\varphi : \mathsf{Group}_1(G, H)$ is sent to $\mathbf{B}\varphi : \mathbf{B}G \to \mathbf{B}H$, as in Definition 14. On 2-cells, let $r : \mathsf{Group}_2(\varphi, \psi)$ a conjugator of homomorphisms. Delooping assigns a 2-cell $\mathbf{B}\varphi = \mathbf{B}\psi$ as follows: By function extensionality, it suffices to give $\mathbf{B}\varphi(x) = \mathbf{B}\psi(x)$ for any $x : \mathbf{B}G$. By induction on x, we are left to give some $q : \bullet = \bullet$ in $\mathbf{B}H$ such that for all $g : G$, $\mathsf{loop}\,\varphi(g) \cdot q = q \cdot \mathsf{loop}\,\psi(g)$. Choose $q := \mathsf{loop}\,r$, and compute

$$\mathsf{loop}\,\varphi(g) \cdot \mathsf{loop}\,r = \mathsf{loop}\,(\varphi(g)r) \overset{(*)}{=} \mathsf{loop}\,(r\psi(g)) = \mathsf{loop}\,r \cdot \mathsf{loop}\,\psi(g),$$

where $(*)$ uses that r is a conjugator of φ and ψ. By a similar argument, one shows that these assignments preserve composition and identities. ◀

Defined this way, we see that $\mathbf{B}$ preserves the local structure of Group:

▶ **Theorem 34.** *The functor* $\mathbf{B} : \mathsf{Group} \to \mathsf{hGpd}$ *is locally a weak equivalence of categories, i.e. for all groups* G, H*, there merely exists an inverse functor* $\mathsf{hGpd}_1(\mathbf{B}G, \mathbf{B}H) \to \mathsf{Group}_1(G, H)$*.*

Note that this cannot be strengthened to a full equivalence with explicit inverse: equivalence of categories preserves properties, but hGpd is by definition locally univalent, whereas Group is not.

We prove Theorem 34 by showing that locally, $\mathbf{B}$ is fully faithful and essentially surjective.

▶ **Proposition 35.** *Delooping is a locally fully faithful functor: For groups G, H, the local functor $\mathbf{B} : \mathsf{Group}_1(G, H) \to \mathsf{hGpd}_1(\mathbf{B}G, \mathbf{B}H)$ is an equivalence of categories.*

Proof. Let $\varphi, \psi : \mathsf{Group}_1(G, H)$. We establish a chain of equivalences between the sets of 2-cells $\mathsf{Group}_2(\varphi, \psi)$ and $\mathbf{B}\varphi = \mathbf{B}\psi$. Starting from the definition,

$$\mathsf{Group}_2(\varphi, \psi) \simeq \sum_{h:H} \prod_{g:G} \varphi(g)h = h\psi(g)$$

we apply the equivalence of groups, $\mathsf{loop} : H \simeq \Omega(\mathbf{B}H)$ twice

$$\simeq \sum_{h:H} \prod_{g:G} \mathsf{loop}\,\varphi(g) \cdot \mathsf{loop}\,h = \mathsf{loop}\,h \cdot \mathsf{loop}\,\psi(g)$$

$$\simeq \sum_{\ell:\Omega(\mathbf{B}H)} \prod_{g:G} \mathsf{loop}\,\varphi(g) \cdot \ell = \ell \cdot \mathsf{loop}\,\psi(g)$$

By the recursion principle, this is exactly a type of dependent functions out of $\mathbf{B}G$, namely

$$\simeq \prod_{x:\mathbf{B}G} \mathbf{B}\varphi(x) = \mathbf{B}\psi(x)$$

which, by function extensionality, is equivalent to

$$\simeq \mathbf{B}\varphi = \mathbf{B}\psi$$

One verifies that the map underlying this chain is that of Lemma 33. ◀

▶ **Proposition 36.** *Delooping is a locally essentially surjective functor.*

Proof. Let G, H groups, and $f : \mathbf{B}G \to \mathbf{B}H$ a morphism of groupoids. We show the *mere* existence of some $\varphi : \mathsf{Group}_1(G, H)$ together with an isomorphism $\mathbf{B}\varphi \cong f$ in the local category $\mathsf{hGpd}(\mathbf{B}G, \mathbf{B}H)$. By definition, morphisms in this category are homotopies, and it suffices to exhibit some $h : \prod_{x:\mathbf{B}G} \mathbf{B}\varphi(x) = f(x)$. Since $\mathbf{B}H$ is a *connected* groupoid, there merely exists a path $p : f(\bullet) = \bullet$, and conjugation by p induces an equivalence of groups,

$$\begin{aligned} \mathsf{conj}(p) &: \Omega(\mathbf{B}H, f(\bullet)) \to \Omega(\mathbf{B}H, \bullet) \\ &\quad (q : f(\bullet) = f(\bullet)) \mapsto p^{-1} \cdot q \cdot p \end{aligned}$$

We define φ to be the composite

$$G \xrightarrow{\mathsf{loop}} \Omega(\mathbf{B}G, \bullet) \xrightarrow{\mathsf{cong}(f)} \Omega(\mathbf{B}H, f(\bullet)) \xrightarrow{\mathsf{conj}(p)} \Omega(\mathbf{B}H, \bullet) \xrightarrow{\mathsf{loop}^{-1}} H$$

By induction on $x : \mathbf{B}G$, we show that $\prod_{x:\mathbf{B}G} \mathbf{B}\varphi(x) = f(x)$. On the point, this is given by $p^{-1} : \bullet = f(\bullet)$. On loops, we construct $\prod_{g:G} \mathbf{B}\varphi(\mathsf{loop}\,g) \cdot p^{-1} = p^{-1} \cdot f(\mathsf{loop}\,g)$ as follows: let $g : G$, then $\mathbf{B}\varphi(\mathsf{loop}\,g) \cdot p^{-1} = \mathsf{loop}\,(\mathsf{loop}^{-1}\,(p^{-1} \cdot f(\mathsf{loop}\,g) \cdot p)) \cdot p^{-1} = p^{-1} \cdot f(\mathsf{loop}\,g)$ ◀

4.2 A 2-Category of Group Actions

Any G-action σ comes with an associated bundle, $\bar{\mathbf{B}}\sigma : \mathsf{loop}\, G \to \mathsf{hSet}$ (Definition 15). Let us define 2-categories of actions and of set bundles, and show that “taking the associated bundle” is a functorial construction.

▶ **Definition 37** (2-category of actions). *The 2-category* Action *of group actions displayed over* Group *consists of the following data:*

- *For each group G, objects are G-actions* $\mathsf{Action}_0(G) := \sum_{X:\mathsf{hSet}} G \dashrightarrow \mathfrak{S}(X)$.
- *1-cells between actions* $(X,\sigma) : \mathsf{Action}_0(G)$ *and* $(Y,\tau) : \mathsf{Action}_0(H)$ *over* $\varphi : \mathsf{Group}_1(G,H)$ *are equivariant maps* $\mathsf{Action}_1(\varphi; (G,\sigma), (H,\tau)) := \sum_{f:X\leftarrow Y} \mathsf{isEquivariant}_{\varphi,f}(\sigma,\tau)$, *where* isEquivariant *is as in Definition 26.*
- *The type of 2-cells between* $f : \mathsf{Action}_1(\varphi; (X,\sigma), (Y,\tau))$ *and* $g : \mathsf{Action}_1(\psi; (X,\sigma), (Y,\tau))$ *over a conjugator* $r : \mathsf{Group}_2(\varphi,\psi)$ *is* $\mathsf{Action}_2(r; f, g) := (f = g \circ \tau(r))$. ⌟

The type of 2-cells says that 1-cells agree up to a permutation of their domain. Since it is a proposition, verifying the axioms of a displayed 2-category reduces to defining *some* identity- and composite 2-cells. The vertical composite $p \bullet q : f \Rightarrow_{rs} h$ of 2-cells $p : f \Rightarrow_r g$ and $q : g \Rightarrow_s h$, is some identification $f = h \circ \tau(rs)$. Since τ is an action, we define the composite as $p \bullet q := \big(f \overset{p}{=} g \circ \tau(r) \overset{q}{=} h \circ \tau(s) \circ \tau(r) = h \circ \tau(rs)\big)$. Similarly, horizontal composition depends on 2-cells of Group being conjugators of group homomorphisms.

▶ **Definition 38** (set bundles). *The 2-category of* set bundles*, displayed over* hGpd*, consists of the following data:*

- *Given* $G : \mathsf{hGpd}_0$*,* set bundles *on G are families* $\mathsf{SetBundle}_0(G) := G \to \mathsf{hSet}$.
- *Over* $\varphi : \mathsf{hGpd}_1(G,H)$*, morphisms from* $X : \mathsf{SetBundle}(G)$ *to* $Y : \mathsf{SetBundle}(H)$ *are dependent functions,* $\mathsf{SetBundle}_1(\varphi; X, Y) := \prod_{g:G} Y(\varphi g) \to X(g)$
- *Displayed 2-cells between* $f : \mathsf{SetBundle}_1(\varphi; X, Y)$ *and* $g : \mathsf{SetBundle}_1(\psi; X, Y)$ *over a 2-cell* $p : \varphi = \psi$ *are dependent identifications,* $\mathsf{SetBundle}_2(p; f, g) := f =_p g$. ⌟

For an object $(G, F) : \mathsf{SetBundle}_0$ in the total category, we call G the *base* of the bundle, and F its fibers.

We are now ready to show that taking the bundle associated to an action is a well-behaved functorial operation. In particular, each equivariant map of actions induces a morphism between associated bundles:

▶ **Definition 39.** *Let* $\sigma : \mathsf{Action}(G,X)$*,* $\tau : \mathsf{Action}(H,Y)$*, and* $\varphi : \mathsf{Group}_1(G,H)$*. Let* $f : Y \leftarrow X$ *and* $p : \mathsf{isEquivariant}_{\varphi,f}(\sigma,\tau)$*. The bundle morphism associated to f has type* $\bar{\mathbf{B}}f : \prod_{x:\mathbf{B}G} \bar{\mathbf{B}}\tau(\mathbf{B}\varphi x) \to \bar{\mathbf{B}}\sigma(x)$*, and is defined using the induction principle of* $\mathbf{B}G$*. On the point it has type* $\bar{\mathbf{B}}f(\bullet) : Y \to X$ *and is given by f. On a loop, we need to prove that* $\bar{\mathbf{B}}\sigma(\mathsf{loop}\, g) \circ f = f \circ \bar{\mathbf{B}}\tau(\mathbf{B}\varphi(\mathsf{loop}\, g))$ *for all* $g : G$*. This reduces to* $\sigma(g) \circ f = f \circ \tau(\varphi(g))$*, which is given by p.*

Both Action and SetBundle are total categories, and $\mathbf{B} : \mathsf{Group} \to \mathsf{hGpd}$ is a 2-functor between their bases. We thus define a 2-functor only on the displayed parts:

▶ **Definition 40.** *Taking associated bundles is a displayed 2-functor* $\bar{\mathbf{B}} : \mathsf{Action} \to_{\mathbf{B}} \mathsf{SetBundle}$*, consisting of the following data:*

- *On objects, it sends a G-action σ to its associated bundle* $\bar{\mathbf{B}}\sigma : \mathbf{B}G \to \mathsf{hSet}$.
- *On 1-cells, it associates to an equivariant map* $f : \mathsf{Action}_1(\varphi; \sigma, \tau)$ *its morphism of bundles* $\bar{\mathbf{B}}f : \mathsf{SetBundle}_1(\mathbf{B}\varphi; \bar{\mathbf{B}}\sigma, \bar{\mathbf{B}}\tau)$.
- *Over a 2-cell* $r : \mathsf{Group}_2(G,H)$*, a proof* $p : \mathsf{Action}_2(r; f, g) \doteq (f = g\tau(r))$ *is sent to a homotopy of bundle maps using the induction principle of* $\mathbf{B}G$.

Both actions on 1- and 2-cells are defined by induction, and thus are equivalences in the sense of Proposition 13:

▶ **Lemma 41.** *The action on 1-cells* $\bar{\mathbf{B}}_1 : \mathsf{Action}_1(\varphi; \sigma, \tau) \to \mathsf{SetBundle}_1(\mathbf{B}\varphi; \bar{\mathbf{B}}\sigma, \bar{\mathbf{B}}\tau)$ *and 2-cells* $\bar{\mathbf{B}}_2 : \mathsf{Action}_2(r; f, g) \to \mathsf{SetBundle}_2(\mathbf{B}r; \bar{\mathbf{B}}f, \bar{\mathbf{B}}g)$ *are equivalences of types.*

▶ **Theorem 42.** *Taking associated bundles* $\int \bar{\mathbf{B}} : \mathsf{Action} \to \mathsf{SetBundle}$ *is locally a weak equivalence.*

Proof. The total functor $\int \bar{\mathbf{B}}$ is locally fully faithful if $\int_2 \bar{\mathbf{B}}$ is an equivalence, but this is a map on Σ-types built from $\mathbf{B}_2$ and $\bar{\mathbf{B}}_2$, which are both equivalences by Theorem 34 and Lemma 41. Local essential surjectivity is proved similarly to Proposition 36, and uses that $\bar{\mathbf{B}}_1$ is an equivalence of types. ◀

The above theorem implies that the local category $\mathsf{SetBundle}(\mathbf{B}G, \mathbf{B}H)$ is the Rezk completion of $\mathsf{Action}(G, H)$. As such the 2-category $\mathsf{SetBundle}$ should be the local univalent completion of Action in the sense of [6, Conjecture 5.6].

4.3 A 2-Categorical Fam-Construction

In the previous section we have seen how containers with a single action relate to set bundles. To lift this relationship to families of actions, we introduce a 2-categorical Fam-construction, again employing displayed machinery.

▶ **Definition 43** (2-category of families). *Let C be a 2-category. The 2-category* $\mathrm{Fam}(C)$ *displayed over* hSet *consists of the following data:*

- *For $J : \mathsf{hSet}_0$, the displayed objects are families of C-objects,* $\mathrm{Fam}_0(J) := J \to C_0$.
- *Let $J, K : \mathsf{hSet}_0$ and families $X : \mathrm{Fam}_0(J)$, $Y : \mathrm{Fam}_0(K)$. The type of 1-cells displayed over some $\varphi : \mathsf{hSet}_1(J, K)$ is* $\mathrm{Fam}_1(\varphi; X, Y) := \prod_{j:J} C_1(X_j, Y_{\varphi j})$.
- *Displayed 2-cells are a family* $\mathrm{Fam}_2 : (\varphi = \psi) \to \mathrm{Fam}_1(\varphi, X, Y) \to \mathrm{Fam}_1(\psi, X, Y) \to \mathcal{U}$ *defined by path-induction on 2-cells in* hSet*:* $\mathrm{Fam}_2(\mathsf{refl}_\varphi; f, g) := \prod_{j:J} C_2(f_j, g_j)$

▶ **Definition 44.** *Any 2-functor $F : \mathsf{C} \to \mathsf{D}$ lifts to a functor* $\mathrm{Fam}(F) : \mathrm{Fam}(\mathsf{C}) \to \mathrm{Fam}(\mathsf{D})$*. This lifting is defined as a total functor* $\mathrm{Fam}(F) : \int_{J:\mathsf{hSet}} \mathrm{Fam}(\mathsf{C})(J) \to \int_{J:\mathsf{hSet}} \mathrm{Fam}(\mathsf{D})(J)$ *over the identity 2-functor on the base* hSet*.*

▶ **Proposition 45.** *Lifting $F : \mathsf{C} \to \mathsf{D}$ to a 2-functor of families inherits the following properties:*

1. *If F is locally fully-faithful, so is* $\mathrm{Fam}(F)$*.*
2. *If F is locally split-essentially surjective, so is* $\mathrm{Fam}(F)$*.*
3. *Assuming the axiom of choice for h-sets and that C is locally strict, if F is locally essentially surjective, so is* $\mathrm{Fam}(F)$*.*

Proof. Local fully-faithfulness follows from the pointwise definition of Fam_2: if $\mathsf{C}_2(f, g)$ is an equivalence, then so is $\mathrm{Fam}_2(\mathsf{refl}; -, -) \doteq \lambda f, g. \prod_j \mathsf{C}_2(f_j, g_j)$. Local split essential surjectivity follows from a similar pointwise argument.

In the non-split case, fix $x, y : \mathrm{Fam}_0(\mathsf{C})$, and a 1-cell $(\psi, g) : \mathrm{Fam}_0(x) \to \mathrm{Fam}_0(y)$. It suffices to provide merely a family of sections, $\|\prod_{j:J} \sum_f F_1(f) \cong g_j\|_{-1}$; the conclusion follows using the induction principle of the truncation. The assumption that F is locally eso yields $\prod_{j:J} \|\sum_{f:\mathsf{C}_1(x_j, y_{\psi j})} F_1(f) \cong g_j\|_{-1}$, and we use choice to move the truncation outward: J is a set, and so are $\mathsf{C}_1(x_j, y_{\psi j})$ (by local strictness of C) and local isomorphisms $F_1(f) \cong g_j$. ◀

4.4 Action Containers as a 2-Category of Families

As promised, we define the 2-category of action containers as a 2-category of families:

▶ **Definition 46** (2-category of action containers)**.** *The 2-category of action containers is that of families of actions,* $\mathsf{ActCont} := \mathrm{Fam}(\mathsf{Action})$.

By algebra of Σ- and Π-types, we see that the objects and 1-cells coincide with those of the 1-category of action containers (cf. Definition 25). In particular, we have for objects

$$\mathsf{ActCont}_0 \simeq \sum_{S:\mathsf{hSet}} \sum_{P:S\to\mathsf{hSet}} \sum_{G:S\to\mathsf{Group}} \prod_{s:S} \mathsf{Action}(G_s, P_s),$$

and for 1-cells,

$$\begin{aligned}&\mathsf{ActCont}_1((S, \lambda s.\,(P_s, G_s, \sigma_s)), (T, \lambda t.\,(Q_t, H_t, \tau_t))) \\ &\quad\simeq \sum_{u:S\to T} \prod_{s:S} \sum_{\varphi:G_s\to H_{us}} \sum_{f:P_s\leftarrow Q_{us}} \mathsf{isEquivariant}_{\varphi,f}(\sigma_s, \tau_{us})\end{aligned}$$

Unfolding the newly added type of 2-cells, we recover a familiar condition:

▶ **Proposition 47.** *Let* $E, F : \mathsf{ActCont}_0$ *and denote* $E \doteq (S, \lambda s.\,(P_s, G_s, \sigma_s))$ *and* $F \doteq (T, \lambda t.\,(Q_t, H_t, \tau_t))$. *Let* $u : S \to T$ *and* $f, g : \mathsf{ActCont}_1(u; E, F)$. *The type of 2-cells* $\mathsf{ActCont}_2((u, f), (u, g))$ *is equivalent to*

$$\prod_{s:S} \sum_{r:H_{us}} \mathsf{isConjugator}_{\varphi_s,\psi_s}(r) \times f'_s = g'_s \circ \tau_{us}(r),$$

in which $\varphi, \psi : \prod_s G_s \dot{\to} H_{us}$ *and* $f', g' : \prod_s Q_{us} \to P_s$ *are the maps of symmetries and positions of* f *and* g*, respectively.*

Note the occurrence of the proposition $f'_s = g'_s \circ \tau_{us}(r)$: it has already appeared in the definition of quotient container morphisms as a quotient of premorphisms (Definition 6). In [3, Definition 4.1] it is explained as a necessary condition for labellings of container maps to be defined upto quotient. In our case it is necessary to characterize homotopies between induced bundle maps $\bar{\mathbf{B}}_1(f'_s)$ and $\bar{\mathbf{B}}_1(g'_s)$ (Lemma 41).

Lifting the 2-functor $\bar{\mathbf{B}} : \mathsf{Action} \to \mathsf{SetBundle}$ to families, we immediately obtain the following characterization of 1-cells of action containers. Substituting $\mathsf{ActCont} \doteq \mathrm{Fam}(\mathsf{SetBundle})$ and $\mathsf{SymmCont} \doteq \mathsf{SetBundle}$, we see:

▶ **Corollary 48.** *The lifting* $\mathrm{Fam}(\bar{\mathbf{B}}) : \mathsf{ActCont} \to \mathrm{Fam}(\mathsf{SymmCont})$ *is:*

1. *locally fully faithful*
2. *assuming the axiom of choice, locally a weak equivalence*

Proof. By application of Proposition 45: The 2-category Action is locally strict, and $\bar{\mathbf{B}}$ locally a weak equivalence (Theorem 42). ◀

This says that constructively, morphisms of action containers correspond to a subcategory of morphisms of (families of) symmetric containers. By using the axiom of choice, one sees that this construction indeed covers all such morphism.

To connect action containers to symmetric ones, and not just families of the latter, note the following: Any family of set bundles (hence symmetric containers) can be combined into a single bundle: given $(J, (\lambda j.\,(B_j, F_j))) : \mathrm{Fam}(\mathsf{SetBundle})$, we can consider the bundle of fibers over the sum of bases, $\sum_{j:J} B_j$. This construction defines a 2-functor:

▶ **Definition 49** (summation of set bundles). *Summation of set bundles is a 2-functor* $\Sigma : \mathrm{Fam}(\mathsf{SetBundle}) \to \mathsf{SetBundle}$*, with the following data*

1. $\Sigma_0(J, \lambda j.\,(B_j, F_j))$ *consists of the base* $\sum_{j:J} B_j$*, and fibers* $\lambda(j,b).\,F_j(b)$*.*
2. $\Sigma_1(u, \lambda j.\,(\varphi_j, f_j))$ *is a pair of a reindexing function* $(u, \varphi_{\_}) : \sum_J B \to \sum_K C$ *and a map of fibers,* $f_{\_} : \prod_{(j,b)} G_{uj}(\varphi_j(b)) \to F_j(b)$*.*
3. *On 2-cells, it takes a family of identities of bundle maps to an identity of their sums via function extensionality:* $\Sigma_2(\mathsf{refl}_u, \lambda j.\,(r_j, \bar{r}_j)) :\equiv \mathsf{funext}\,\lambda(j,b).\,\mathsf{cong}_{uj,\text{-}b}(r_j), \bar{r}_j(b)$*.* ⌟

The construction turning action containers into symmetric ones now factors as follows:

$$\mathsf{ActCont} \xrightarrow{=} \mathrm{Fam}(\mathsf{Action}) \xrightarrow{\mathrm{Fam}(\bar{\mathbf{B}})} \mathrm{Fam}(\mathsf{SetBundle}) \xrightarrow{\Sigma} \mathsf{SetBundle} \xrightarrow{=} \mathsf{SymmCont}$$

In general, we do not know whether Σ is locally fully faithful or not. We can however consider its restriction to objects in the image of $\mathrm{Fam}(\bar{\mathbf{B}})$, and deduce fully-faithfulness for some of its local functors:

▶ **Lemma 50.** *Let* $X, Y : \mathrm{Fam}_0(\mathsf{SetBundle})$*. If all bundles in* X *have connected bases, then the local functor* $\Sigma_1 : \mathrm{Fam}_1(X, Y) \to \mathsf{SetBundle}(\Sigma_0(X), \Sigma_0(Y))$ *is fully faithful.*

Proof. The proof proceeds by showing that there is an equivalence of 2-cells. The assumption on connectedness is used as follows: Recall that $X \doteq (J, \lambda j.\,(B_j, F_j))$ has connected bases if all B_j are connected groupoids. The base of the bundle $\Sigma_0(X)$ is $\sum_j B_j$, and maps between such bases are typed $\sum_j B_j \to \sum_k B'_k$. To characterize identifications of these maps, it is necessary show, given morphisms $u, v : J \to K$, that the function $u(j) = v(j) \to (B_j \to u(j) = v(j))$ constant in B_j is an equivalence. But this follows from connectedness of B_j and the fact that $u(j) = v(j)$ is a proposition [20, 7.5.9]. ◀

▶ **Theorem 51.** *The composite* $\Sigma \circ \mathrm{Fam}(\bar{\mathbf{B}}) : \mathsf{ActCont} \to \mathsf{SymmCont}$ *is locally fully faithful.*

5 Conclusions

We introduced action containers for studying data types with symmetries. This class of containers is inspired by quotient containers, but are different from the latter in two key aspects: First, symmetries are encoded by arbitrary groups acting on positions, allowing for more freedom in presenting permutations of positions. Second, morphisms are required to respect symmetries in a coherent way, and are additionally not equivalence classes of an ad-hoc relation. Instead, the category of action containers is presented universally as a free coproduct completion of a category of groups and actions, from which limits and colimits of action containers are easy to read off. We reintroduce the relation between morphisms in terms of a 2-categorical structure, and show that the 2-category of action containers embeds into that of symmetric containers.

A missing piece in our analysis is the relationship between quotient and action containers. The latter subsume the former, but morphisms of action containers are more constrained than those of quotient containers. Finding a functorial connection is not straightforward: Each action container $(S \triangleright P \triangleleft^{\sigma} G)$ can be mapped to a quotient container with the same set of shapes and positions, but for each $s : S$ changing the group to $\mathsf{Im}(\sigma_s)$, which is a subgroup of $\mathfrak{S}(P_s)$. Unfortunately this operation does not act on morphism $(u \triangleright f \triangleleft \varphi) : (S \triangleright P \triangleleft^{\sigma} G) \to (T \triangleright Q \triangleleft^{\tau} H)$, since group homomorphisms $\varphi_s : G_s \dashrightarrow H_{us}$ do not generally restrict to the image of the actions $\mathsf{Im}(\sigma_s) \dashrightarrow \mathsf{Im}(\tau_{us})$. When restricted to a 1-subcategory of action containers with faithful actions, this construction at least yields an isomorphism-on-objects functor

$\mathsf{ActCont}_{\mathsf{faith}} \to \mathsf{QuotCont}$. In the opposite direction, one could search for a functor between the category QuotCont and the homotopy category of ActCont, i.e. the category with the same objects but with sets of morphisms obtained by set quotienting 1-cells by 2-cells. We have a candidate if we modify Definition 5 of premorphism by turning the existential quantifier in the preservation of symmetries into a dependent sum: send a quotient container $(S \triangleright P/G)$ to the action container with the same set of shapes and positions, but for each $s : S$ changing the group to the free group generated by G_s, which also acts on P_s, though not in a faithful way. This modification shows at least the existence of an action of morphisms. We postpone a deeper investigation in this direction to future work.

In Section 3.1 we analyzed some properties of the category of action containers. Ordinary containers enjoy further properties that make them suitable as a model of data types: Altenkirch et al. [8] show that the category of ordinary containers is cartesian closed, and Abbott et al. [3] construct initial algebras and final coalgebras of containers in one parameter. In the future, we plan to investigate whether action containers also enjoy these properties as well. From previous investigations [14], we know that direct construction of final coalgebras for quotient containers fails constructively. In [5] however, Ahrens et al. show that for any $\mathcal{U}$-valued container (with no restriction on truncation level of shapes or positions), its extension as a polynomial in $\mathcal{U}$ admits a final coalgebra. Since extensions of symmetric containers are $\mathcal{U}$-polynomials as well, we would like to internalize this construction: first, find a symmetric container representing this coalgebra, then investigate if this restricts to the inclusion of action containers. This in particular requires studying the composition-monoidal structure of action containers. Similarly, it should be possible to lift the closure properties of Section 3.1 from the underlying 1-category to proper 2-(co)limits in Action.

In another direction we are interested to see if the heavy machinery of 2-categories can be avoided, or at least be postponed. This is guided by the following insight: The image on objects of the 2-functor $\mathbf{B}$ is not only groupoids, but *pointed, connected groupoids.* The 2-category of such groupoids and pointed maps, displayed over hGpd is, surprisingly, locally thin [9, Lemma 4.4.12]. In other words, pointed, connected groupoids and pointed maps form a 1-category hGroup. A slight modification of the proof of Proposition 36 shows that the 1-category of ordinary groups is equivalent to hGroup, and that this equivalence extends to categories of actions. The same argument shows that the 2-category of *skeletal groupoids*[2] and skeleton-preserving maps is locally thin. We believe that this extends to an equivalence between the 1-categories of action containers (without additional relation between morphisms) and skeletal symmetric containers, i.e. those whose shapes are skeletal groupoids, and whose morphisms preserve such skeleta. This would identify a structure on symmetric containers such that equality of morphisms becomes propositional, giving rise to a convenient 1-category of containers with symmetries.

References

1 Michael Abbott, Thorsten Altenkirch, and Neil Ghani. Categories of Containers. In Andrew D. Gordon, editor, *Proc. of 6th Int. Conf. on Foundations of Software Science and Computation Structures, FoSSaCS'03*, volume 2620 of *Lecture Notes in Computer Science*, pages 23–38. Springer Berlin Heidelberg, 2003. doi:10.1007/3-540-36576-1_2.

2 Michael Abbott, Thorsten Altenkirch, and Neil Ghani. Containers: Constructing strictly positive types. *Theoretical Computer Science*, 342(1):3–27, 2005. doi:10.1016/j.tcs.2005.06.002.

[2] Those that come with a choice of basepoint $\mathsf{pt} : \prod_{x:\|B\|_0} \mathsf{fiber}_{|-|_0}(x)$ for each connected component.

3 Michael Abbott, Thorsten Altenkirch, Neil Ghani, and Conor McBride. Constructing Polymorphic Programs with Quotient Types. In Dexter Kozen and Carron Shankland, editors, *Proc. of 7th Int. Conf. on Mathematics of Program Construction, MPC'04*, volume 3125 of *Lecture Notes in Computer Science*, pages 2–15. Springer Berlin Heidelberg, 2004. `doi:10.1007/978-3-540-27764-4_2`.

4 Jiří Adámek and Jiří Rosický. How nice are free completions of categories? *Topology and its Applications*, 273:106972, 2020. `doi:10.1016/j.topol.2019.106972`.

5 Benedikt Ahrens, Paolo Capriotti, and Régis Spadotti. Non-Wellfounded Trees in Homotopy Type Theory. In Thorsten Altenkirch, editor, *Proc. of 13th Int. Conf. on Typed Lambda Calculi and Applications, TLCA'15*, volume 38 of *LIPIcs*, pages 17–30. Schloss Dagstuhl, 2015. `doi:10.4230/LIPICS.TLCA.2015.17`.

6 Benedikt Ahrens, Dan Frumin, Marco Maggesi, Niccolò Veltri, and Niels van der Weide. Bicategories in univalent foundations. *Mathematical Structures in Computer Science*, 31(10):1232–1269, 2021. `doi:10.1017/s0960129522000032`.

7 Benedikt Ahrens and Peter LeFanu Lumsdaine. Displayed Categories. *Logical Methods in Computer Science*, 15(1), March 2019. `doi:10.23638/lmcs-15(1:20)2019`.

8 Thorsten Altenkirch, Paul Blain Levy, and Sam Staton. Higher-Order Containers. In Fernando Ferreira, Benedikt Löwe, Elvira Mayordomo, and Luís Mendes Gomes, editors, *6th Conf. on Computability in Europe, CiE'10*, volume 6158 of *LNCS*, pages 11–20. Springer, 2010. `doi:10.1007/978-3-642-13962-8_2`.

9 Marc Bezem, Ulrik Buchholtz, Pierre Cagne, Bjørn Ian Dundas, and Daniel R. Grayson. Symmetry. `https://github.com/UniMath/SymmetryBook`. Commit: `8546527`.

10 Håkon Robbestad Gylterud. Symmetric Containers. Master's thesis, Department of Mathematics, Faculty of Mathematics and Natural Sciences, University of Oslo, 2011. URL: `https://hdl.handle.net/10852/10740`.

11 Ryu Hasegawa. Two applications of analytic functors. *Theoretical Computer Science*, 272(1–2):113–175, February 2002. `doi:10.1016/s0304-3975(00)00349-2`.

12 Pieter Hofstra and Martti Karvonen. Inner automorphisms as 2-cells. *Theory and Applications of Categories*, 42(2):19–40, 2024. URL: `http://www.tac.mta.ca/tac/volumes/42/2/42-02abs.html`.

13 Niles Johnson and Donald Yau. *2-Dimensional Categories*. Oxford University Press, January 2021. `doi:10.1093/oso/9780198871378.001.0001`.

14 Philipp Joram and Niccolò Veltri. Constructive Final Semantics of Finite Bags. In Adam Naumowicz and René Thiemann, editors, *Proc. of 14th Int. Conf. on Interactive Theorem Proving, ITP'23*, volume 268 of *LIPIcs*, pages 12:1–12:19. Schloss Dagstuhl, 2023. `doi:10.4230/LIPICS.ITP.2023.20`.

15 André Joyal. Foncteurs analytiques et espèces de structures. In Gilbert Labelle and Pierre Leroux, editors, *Combinatoire énumérative*, volume 1234 of *Lecture Notes in Mathematics*, pages 126–159. Universite du Quebec a Montreal, Springer Berlin Heidelberg, May 1986. `doi:10.1007/bfb0072514`.

16 Daniel R. Licata and Eric Finster. Eilenberg-MacLane spaces in homotopy type theory. In Thomas A. Henzinger and Dale Miller, editors, *Proc. of the Joint Meeting of the 23rd EACSL Ann. Conf. on Computer Science Logic (CSL) and the 29th Ann. ACM/IEEE Symp. on Logic in Computer Science (LICS), CSL-LICS'14*, pages 66:1–66:9. ACM, 2014. `doi:10.1145/2603088.2603153`.

17 Max New, Steven Shaefer, and contributors. `cubical-categorical-logic`, 2024. URL: `https://github.com/maxsnew/cubical-categorical-logic`.

18 Andrew M. Pitts. *Nominal sets*. Number 57 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2013.

19 The `agda/cubical` development team. The `agda/cubical` library, 2018. URL: `https://github.com/agda/cubical/`.

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