International Electronic Journal of Algebra
Volume 34 (2023) 21-30
DOI: 10.24330/ieja.1260475

A NOTE ON A FREE GROUP. THE DECOMPOSITION OF A FREE GROUP FUNCTOR THROUGH THE CATEGORY OF HEAPS

Bernard Rybołowicz

Received: 2 November 2021; Revised: 13 December 2022; Accepted: 17 January 2023
Communicated by Abdullah Harmancı

Abstract. This note aims to introduce a left adjoint functor to the functor which assigns a heap to a group. The adjunction is monadic. It is explained how one can decompose a free group functor through the previously introduced adjoint and employ it to describe a slightly different construction of free groups.

Mathematics Subject Classification (2020): 97H40, 20N10, 08A62, 08B20
Keywords: Heap, free group, free group functor

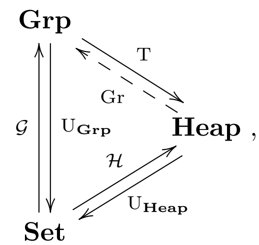
1. Introduction

In the construction of a free group (see [6, Chapter 6] and [7]), one extends a generating set by a neutral element and inverse elements. The method of adding those elements happen entirely by using the set-theoretic operation, i.e. a disjoint union of sets. The aim of this article is to suggest an approach towards free groups that exhibits an algebraic interpretation of the set-theoretic operation of extending the generating set by identity and inverses. To give those procedures new algebraic meaning, we use heaps, a specific variant of universal algebras introduced by R. Baer [1] and H. Prüfer [9]. Due to both Prüfer [9] (Abelian case) and R. Baer [1] (general case), we know that with every heap, we can associate a group, and with every group, we can assign a heap. The latter is a functor.

This paper aims to construct a left adjoint to the functor $\mathrm{T} : \mathbf{Grp} \to \mathbf{Heap}$ between categories of groups and heaps. In the case of a non-empty heap $(H, [-,-,-])$, we can assign to every element $e \in H$ a group $\mathrm{G}(H; e)$. This assignment is not functorial due to the lack of choice of an element for the empty heap. In Theorem 3.5, we overcome this problem by taking a coproduct of a heap with a singleton heap. A singleton heap is a free object over a singleton set in the category of heaps. We observe that taking the coproduct gives us the universal property, see Lemma 3.1.

This work was supported by the EPSRC research grant EP/V008129/1 and Swansea University Research Excellence Scholarships (SURES).

That allows us to construct a functor from heaps to groups, i.e. it lets us fix added elements on morphisms. That leads us to the following diagram



where $\mathrm{U}_{\mathbf{Grp}}, \mathrm{U}_{\mathbf{Heap}}$ are forgetful functors, $\mathcal{G}$ and $\mathcal{H}$ are free groups and heap functors, respectively, T is the assignment of a heap to a group, and Gr is a functor from Theorem 3.5. It is possible to use the adjoint lifting theorem to provide the existence of the left adjoint Gr of T, which completes the diagram. However, in this paper, we present an explicit construction of Gr without using the free group functor, which enables us to provide an alternative explicit construction of the free group, where the aforementioned set-theoretic operations are rendered more explicitly as algebraic operations. In the conclusion part, we compare both constructions of a free group.

2. Preliminaries

Following Baer [1] and Prüfer [9] a *heap* is a set H together with a ternary operation $[-,-,-] : H \times H \times H \to H$ such that for all $h_1, h_2, h_3, h_4, h_5 \in H$ the following holds

$$[[h_1, h_2, h_3], h_4, h_5] = [h_1, h_2, [h_3, h_4, h_5]], \tag{1}$$

$$[h_1, h_2, h_2] = h_1 = [h_2, h_2, h_1]. \tag{2}$$

Equation (1) is called associativity, and equations (2) are called the Malcev identities. If for all $h_1, h_2, h_3 \in H$, $[h_1, h_2, h_3] = [h_3, h_2, h_1]$ we say that H is Abelian.

A *sub-heap* S of a heap H is a subset of H closed under the ternary operation.

A *homomorphism of heaps* is a map between heaps which preserves the ternary operation. Observe that a constant map between two heaps is a heap homomorphism since, by the Malcev identities (2), a single element $e \in H$ is a sub-heap of H.

A sub-heap S is said to be *normal* if there exists $e \in S$ such that for all $h \in H$ and $s \in S$ there exists $s' \in S$ such that

$$[h, e, s] = [s', e, h] \text{ or equivalently } [[h, e, s], h, e] = s'.$$

If S is a normal sub-heap then the quotient heap H/S is well defined and canonical map $\pi : H \to H/S$ is a heap epimorphism, see [4, Proposition 2.10].

An important property of heaps is that with every heap H we can associate a group by choosing an element $e \in H$ and defining a binary operation $+_e := [-, e, -] : H \times H \to H$, we will call the group $(H, +_e)$ *a retract of H in e* or *e-retract* and denote by $\mathrm{G}(H; e)$. It is worth to mention that an assignment G is not a functor, as it is not well-defined on the morphisms. If φ is a homomorphism of heaps then φ is a homomorphism of appropriate retracts if and only if it preserves neutral elements of the retracts.

In the opposite direction, one can associate with every group G a heap by defining ternary operation, for all $a, b, c \in G$, as $[a, b, c] := ab^{-1}c$. We call this heap a *heap associated with a group G* and denote it by $\mathrm{T}(G)$. In contrast to the assignment G, the assignment $\mathrm{T} : \mathbf{Grp} \to \mathbf{Heap}$, between categories of groups and heaps, is a functor given on morphisms $\varphi : G \to G'$ by $\mathrm{T}(\varphi) = \varphi$. Every group homomorphism is a homomorphism of associated heaps.

By employing both assignments to a group G one gets that a group $\mathrm{G}(\mathrm{T}(G); e)$ is isomorphic to G, for all $e \in G$. By applying assignments to a heap H, we get that $\mathrm{T}(\mathrm{G}(H; e)) = H$, for all $e \in H$. The second link between groups and heaps can be used to show that for all $h_1, h_2, h_3, h_4, h_5 \in H$,

$$[[h_1, h_2, h_3], h_4, h_5] = [h_1, [h_4, h_3, h_2], h_5] = [h_1, h_2, [h_3, h_4, h_5]], \tag{3}$$

see Lemma 2.3 of [4].

The general construction of free objects in varieties of algebras is in [3, Theorem 9.3.3]. A more detailed description of the case of heaps is in [5, Section 3]. Let us briefly describe the construction. Starting with a set X, we define a set of odd-length words constructed from the set X by

$$\mathrm{W}(X) := \{x_1 x_2 \dots x_{2i+1} \mid x_i \in X \ \&\ i \in \mathbb{N}\}.$$

On that set, we define a ternary operation as follows, for any three words $w_1, w_2, w_3 \in \mathrm{W}(X)$,

$$[w_1, w_2, w_3] := w_1 w_2^{\circ} w_3,$$

where $w_1 w_2$ is a juxtaposition of words and $w_2^{\circ} := x_{2i+1} x_{2i} \dots x_1$ for any word $w_2 = x_1 x_2 \dots x_{2i+1}$. Since all the words have odd lengths and the number of letters of $[w_1, w_2, w_3]$ is the sum of letters of words w_1, w_2 and w_3, we get that the operation is well-defined and $[w_1, w_2, w_3] \in \mathrm{W}(X)$. To acquire the Malcev identities, one needs to consider the following relation on the $\mathrm{W}(X)$, for any two words $w_1, w_2 \in \mathrm{W}(X)$

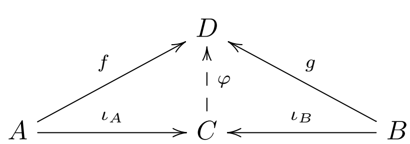
such that $w_1 = x_1 x_2 \dots x_{2i+1}$

$$w_1 \sim w_2 \iff w_2 = x_1 x_2 \dots x_k \hat{x}\hat{x} x_{k+1} x_{2i+1} \qquad \hat{x} \in \mathrm{W}(X).$$

The equivalence relation $\langle\sim\rangle$ generated by the relation $\sim$ makes $\mathcal{H}(X) := (\mathrm{W}(X)/\langle\sim\rangle, [-,-,-])$ a free heap. Any function $f : X \to H$, from the set X to any heap H has a unique extension to the heap homomorphism $\widehat{f} : \mathcal{H}(X) \to H$ and makes $\mathcal{H}$ a functor by taking $\mathcal{H}(h) := \widehat{\iota_Y \circ h} : \mathcal{H}(X) \to \mathcal{H}(Y)$, where $h : X \to Y$, and $\iota : Y \to \mathcal{H}(Y)$ is a canonical injection. We will denote the free heap functor by $\mathcal{H}$: **Set** $\to$ **Heap** to be coherent with the notation of free heaps. It is a left adjoint to the forgetful functor $\mathrm{U}_{\mathbf{Heap}}$: **Heap** $\to$ **Set**. Observe that in this process, we do not enhance the set X of generators of $\mathrm{W}(X)$ by any new letters as it happens in the case of free groups where one needs to add inverses and a neutral element.

Since heaps form a variety of algebras small colimits in **Heap** exist (see [3, Theorem 9.4.14]). For our purposes we are particularly interested in coproducts.

A *coproduct* of two objects A and B in a category $\mathfrak{C}$ is an object C with two morphisms $\iota_A : A \to C$ and $\iota_B : B \to C$, called canonical injections, such that for any object D and morphisms $f : A \to D$ and $g : B \to D$, there exists a unique morphism $\varphi : C \to D$ such that $\varphi \circ \iota_A = f$ and $\varphi \circ \iota_B = g$, i.e. such that the diagram



(4)

commutes.

In Section 3 of [5], one can find the construction of a coproduct in the *category of Abelian heaps* **Ah**, a full subcategory of **Heap**. The idea is to take two Abelian heaps A and B, consider the free heap over theirs disjoint union $\mathcal{H}(A \sqcup B)$ and then divide it by the normal sub-heap generated by

$$[[a,a',a''],[a,a',a'']_A,e], \quad [[b,b',b''],[b,b',b'']_B,e], \tag{5}$$

for all $a,a',a'' \in A$, $b,b',b'' \in B$, where $[---],[---]_A,[---]_B$ are ternary operations in $\mathcal{H}(A \sqcup B)$, A and B, respectively.

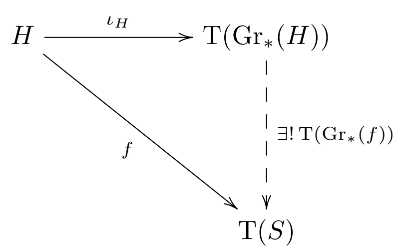
3. Main part

Let us fix some notation. Following [5] we will denote the coproduct of two not necessarily Abelian heaps H and S by $H \boxplus S$. It exists by [3, Theorem 9.4.14], though the author focuses on the more general class of algebras and does not include the explicit description for heaps. We can find the direct description of the coproduct of Abelian heaps in [5, Section 3]. For our consideration, it is enough to know that a free functor preserves small coproducts. The unique filler of the coproduct diagram for morphisms f and g is called a *coproduct map* and is denoted by $f \boxplus g$.

Our main goal is to construct a left adjoint functor to the functor $\mathrm{T} : \mathbf{Grp} \to \mathbf{Heap}$.

A singleton heap is a heap that has only one element, we will denote it by $\{*\}$. For any heap H, one can consider a group $\mathrm{Gr}_*(H) := \mathrm{G}(H \boxplus \{*\}; *)$. The following lemma shows that this group has a very interesting universal property, which will be essential in the construction of the adjoint.

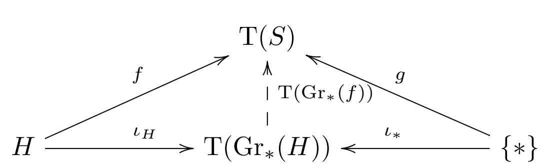
Lemma 3.1. *Let H be a heap, S be a group and $f : H \to \mathrm{T}(S)$ be a heap homomorphism. Then there exists a unique group homomorphism $\mathrm{Gr}_*(f) : \mathrm{Gr}_*(H) \to S$ such that $f = \mathrm{T}(\mathrm{Gr}_*(f)) \circ \iota_H$, where ι_H is a canonical injection into coproduct. In other words, diagram*



(6)

commutes, where $\exists!\,\mathrm{T}(\mathrm{Gr}_*(f))$ reads "there exists exactly one ***homomorphism of groups*** *$\mathrm{Gr}_*(f)$". The pair $(\mathrm{Gr}_*(H), \iota_H)$ is a universal arrow, see* [8, Section III.1].

Proof. Observe that by the universal property of coproduct for all groups S and homomorphisms of heaps $f : H \to \mathrm{T}(S)$ and $g : \{*\} \to \mathrm{T}(S)$ there exists $\mathrm{T}(\mathrm{Gr}_*(f))$ such that the diagram



(7)

commutes. Every homomorphism of groups is a homomorphism of associated heaps. Moreover, a homomorphism of heaps is a homomorphism of retracts if and only if it maps a neutral element to a neutral element. Hence, $\mathrm{T}(\mathrm{Gr}_*(f))$ is a homomorphism of retracts if and only if $g(\iota_*(*))$ is a neutral element of S. Observe that g is unique, since $\{*\}$ is a singleton heap. Therefore $\mathrm{T}(\mathrm{Gr}_*(f))$ is a unique homomorphism of heaps such that it is also a homomorphism of groups to which heaps were associated. Thus, the preceding diagram commutes. □

Another important observation is that a canonical injection ι_H has some sort of cancellation property.

Lemma 3.2. *Let H, L be heaps and $f, g : \mathrm{T}(\mathrm{Gr}_*(H)) \to L$ be homomorphisms of heaps such that $f(\iota_*(*)) = g(\iota_*(*))$, then $f \circ \iota_H = g \circ \iota_H$ implies $f = g$.*

Proof. Let us consider a homomorphism of heaps $f : \mathrm{T}(\mathrm{Gr}_*(H)) \to L$. One can easily observe that by the uniqueness of a coproduct map $f = (f \circ \iota_H) \boxplus (f \circ \iota_*)$. Thus, because $f(\iota_*(*)) = g(\iota_*(*))$ and $f \circ \iota_H = g \circ \iota_H$, we get that

$$f = (f \circ \iota_H) \boxplus (f \circ \iota_*) = (g \circ \iota_H) \boxplus (g \circ \iota_*) = g.$$

□

Corollary 3.3. *Let $e \in L$. If $f, g : \mathrm{Gr}_*(H) \to \mathrm{G}(L; e)$ are homomorphisms of groups, then $f \circ \iota_H = g \circ \iota_H$ implies $f = g$.*

Proof. This follows by Lemma 3.2 since a homomorphism of heaps $\mathrm{T}(f)$ is equal to a homomorphism of groups f as functions. □

Now, we are ready to describe the functor. Let us consider an assignment $\mathrm{Gr} : \mathbf{Heap} \to \mathbf{Grp}$ given on a heap H by $H \mapsto \mathrm{Gr}_*(H)$. One can easily see that it is a well-defined function. The assignment is given for all homomorphisms of heaps $f : H \to H'$ by $f \mapsto \mathrm{Gr}_*(\iota_{H'} \circ f)$. The assignment on morphisms is well-defined since $\iota'_H \circ f$ is a composition of homomorphisms of heaps, so it is a homomorphism of heaps. Therefore by the universal property of Gr_*, $\mathrm{Gr}_*(\iota_{H'} \circ f)$ is a homomorphism of groups.

The following lemma and theorem follows by Theorem 2 (ii) and Theorem 2 (i) in [8, Section IV], respectively. We provide direct proofs for the sake of clarity.

Lemma 3.4. *The assignment* $\mathrm{Gr} : \mathbf{Heap} \to \mathbf{Grp}$ *is a functor.*

Proof. In the previous discussion, we explained that both assignments are well-defined functions. Thus, we have to show that Gr_* preserves identities and composition.

Obviously $\mathrm{Gr}_*(\iota_H \circ id_H) = id_{\mathrm{Gr}_*(H)}$.

For the composition let us assume that $f : H \to H'$ is a homomorphism of heaps, then $\iota_{H'} \circ f$ is a composition of homomorphisms of heaps, hence $\iota_{H'} \circ f : H \to \mathrm{T}(\mathrm{Gr}(H')))$ is a homomorphism of heaps. If $f : H \to H'$ and $h : H' \to H''$

are homomorphisms of heaps, then

$$\begin{aligned}\mathrm{Gr}(h)\circ\mathrm{Gr}(f)\circ\iota_H &= \mathrm{Gr}(h)\circ\mathrm{Gr}_*(\iota_{H'}\circ f)\circ\iota_H = \mathrm{Gr}(h)\circ\mathrm{T}(\mathrm{Gr}_*(\iota_{H'}\circ f))\circ\iota_H\\ &= \mathrm{Gr}(h)\circ\iota_{H'}\circ f = \mathrm{T}(\mathrm{Gr}_*(\iota_{H''}\circ h))\circ\iota_{H'}\circ f = \iota_{H''}\circ h\circ f\\ &= \mathrm{T}(\mathrm{Gr}_*(\iota_{H''}\circ h\circ f))\circ\iota_H = \mathrm{T}(\mathrm{Gr}(h\circ f))\circ\iota_H = \mathrm{Gr}(h\circ f)\circ\iota_H,\end{aligned}$$

where the second, fourth and eighth equality follows from the fact that $\mathrm{T}(g) = g$ for all group homomorphisms g, while the third and fifth follows by the universal property of Gr_*, the sixth is a consequence of both combined. Now, since $\mathrm{Gr}(g\circ f)\circ\iota_H = \mathrm{Gr}(g)\circ\mathrm{Gr}(f)\circ\iota_H$ and $\mathrm{Gr}(g\circ f), \mathrm{Gr}(g), \mathrm{Gr}(f)$ are homomorphisms of groups, applying Corollary 3.3, one gets that $\mathrm{Gr}(g\circ f) = \mathrm{Gr}(g)\circ\mathrm{Gr}(f)$. Therefore an assignment Gr preserves composition, hence Gr is a functor. □

The following theorem confirms that Gr is a desirable functor.

Theorem 3.5. *The functor* Gr *is the left adjoint to the functor* T.

Proof. For all heaps H and groups G let us consider functions between sets of morphisms:

$$\varphi_{H,G} : \mathbf{Grp}(\mathrm{Gr}(H), G) \longrightarrow \mathbf{Heap}(H, \mathrm{T}(G)),\quad g \longmapsto \mathrm{T}(g)\circ\iota_H,$$

$$\varphi^{-1}_{H,G} : \mathbf{Heap}(H, \mathrm{T}(G)) \longrightarrow \mathbf{Grp}(\mathrm{Gr}(H), G),\quad f \mapsto \mathrm{Gr}_*(f).$$

To show that $\varphi_{H,G}$ is a bijection let $f \in \mathbf{Heap}(H, \mathrm{T}(G))$ and $g \in \mathbf{Grp}(\mathrm{Gr}(H), G)$, then

$$\varphi_{H,G}\circ\varphi^{-1}_{H,G}(f) = \varphi_{H,G}(\mathrm{Gr}_*(f)) = \mathrm{T}(\mathrm{Gr}_*(f))\circ\iota_H = f,$$

where the last equality follows by Lemma 3.1, and

$$\varphi^{-1}_{H,G}\circ\varphi_{H,G}(g) = \mathrm{Gr}_*(\mathrm{T}(g)\circ\iota_H) = g,$$

where the last equality follows by the uniqueness of the morphism $\mathrm{Gr}_*(\mathrm{T}(g)\circ\iota_H)$. Hence, $\varphi^{-1}_{H,G}$ is an inverse to $\varphi_{H,G}$. Thus, $\varphi_{H,G}$ is a bijection.

To check the naturality conditions, let G, S be groups, H, L be heaps and consider homomorphisms $f : L \longrightarrow H$ and $g : G \longrightarrow S$. To prove that the φ is natural we need to show that the following diagram

$$\begin{array}{ccc} \mathbf{Grp}(\mathrm{Gr}(H), G) & \xrightarrow{\varphi_{H,G}} & \mathbf{Heap}(H, \mathrm{T}(G)) \\ {\scriptstyle \mathbf{Grp}(\mathrm{Gr}(f),g)}\downarrow & & \downarrow{\scriptstyle \mathbf{Heap}(f,\mathrm{T}(g))} \\ \mathbf{Grp}(\mathrm{Gr}(L), S) & \xrightarrow{\varphi_{L,S}} & \mathbf{Heap}(L, \mathrm{T}(S)) \end{array} \tag{8}$$

commutes, where for all $s\colon \mathrm{Gr}(H) \longrightarrow G$ and $t \in \mathbf{Heap}(H, \mathrm{T}(G))$, $\mathbf{Grp}(\mathrm{Gr}(f), g)(s) := g \circ s \circ \mathrm{Gr}(f)$ and $\mathbf{Heap}(f, \mathrm{T}(g))(t) := \mathrm{T}(g) \circ t \circ f$. Let $s : \mathrm{Gr}(H) \longrightarrow G$, then

$$\begin{aligned}\varphi_{L,G}(g \circ s \circ \mathrm{Gr}(f)) &= \mathrm{T}(g \circ s \circ \mathrm{Gr}(f)) \circ \iota_L = \mathrm{T}(g) \circ \mathrm{T}(s) \circ \mathrm{T}(\mathrm{Gr}(f)) \circ \iota_L \\ &= \mathrm{T}(g) \circ \mathrm{T}(s) \circ \mathrm{T}(\mathrm{Gr}_*(\iota_H \circ f)) \circ \iota_L = \mathrm{T}(g) \circ \mathrm{T}(s) \circ \iota_H \circ f \\ &= \mathrm{T}(g) \circ \varphi_{H,G}(s) \circ f,\end{aligned}$$

where the first implication follows from the definition of $\varphi_{H,G}$, the second follows by the fact that T is a functor and the fourth is the universal property of Gr_*. Therefore φ is a natural isomorphism and the functor Gr is a left adjoint to the functor T. □

Proposition 3.6. *The adjunction* $\mathrm{Gr} \dashv \mathrm{T}$ *is monadic.*

Proof. By [2, 3.14. Theorem] it is enough to show that T reflects isomorphisms, **Grp** has coequalizers of T-split parallel pairs, and T preserves those coequalizers.

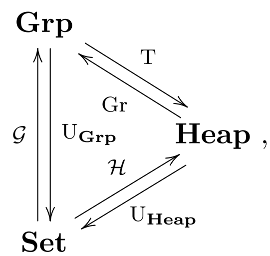
Let us start with the property of reflecting isomorphisms. Let $g : G \to G'$ be a homomorphism of groups such that $\mathrm{T}(g)$ is an isomorphism of heaps. Then, $\mathrm{T}(g)^{-1}$ provides an inverse of g as functions. The fact that $\mathrm{T}(g)^{-1}$ is a group homomorphism follows from the fact that g is a group homomorphism. Therefore T reflects isomorphisms.

To prove that coequalizers for T-split parallel pairs exist let $s, g : G \to G'$ be an T-split parallel pair, then by definition there exist heap H and heap homomorphisms

$$h : \mathrm{T}(G') \to H,\ t : H \to \mathrm{T}(G'),\ f : \mathrm{T}(G') \to \mathrm{T}(G)$$

such that $h \circ \mathrm{T}(s) = h \circ \mathrm{T}(g)$, f and t are sections of $\mathrm{T}(s)$ and h, respectively, and $\mathrm{T}(g) \circ f = t \circ h$. One can check that existence of those homomorphisms imply that a pair (H, h) is a coequalizer of $\mathrm{T}(s)$ and $\mathrm{T}(g)$. Now, let us denote by $e \in G$ and $e' \in G'$ the neutral elements of the groups, then we can consider an $h(e')$-retract of H, a group $\mathrm{G}(H; h(e'))$. Observe that all the aforementioned homomorphisms preserve the neutral elements of groups. Thus, all of those homomorphisms are homomorphisms of the appropriate retracts. Therefore $(\mathrm{G}(H, e), h)$ is a coequalizer of T-split parallel pair (s, g) and h is surjective. Thus, $\mathrm{T}(\mathrm{G}(H; h(e'))) = H$ and T preserves the coequalizers. Hence, the adjunction is monadic. □

To underline the meaning of the preceding theorem in the context of groups let us consider the following diagram



where $U_{\mathbf{Grp}}$ is a forgetful functor and $\mathcal{G}$ is its left adjoint, the free functor.

The first observation is that all the opposite arrows are adjoints to each other.

The second observation is that the composition of functors

$$\mathbf{Grp} \xrightarrow{\mathrm{T}} \mathbf{Heap} \xrightarrow{\mathrm{U}_{\mathbf{Heap}}} \mathbf{Set}$$

is a forgetful functor $\mathrm{U}_{\mathbf{Grp}}$ since for any group G, $\mathrm{T}(G)$ and G are equal sets, and every homomorphism of groups f is the same function as $\mathrm{T}(f)$. These two observations leads to the following corollaries.

Corollary 3.7. *A functor* $\mathrm{Gr} \circ \mathcal{H} : \mathbf{Set} \to \mathbf{Grp}$ *is a free functor, i.e. it is the left adjoint to the functor* $\mathrm{U}_{\mathbf{Grp}} : \mathbf{Grp} \to \mathbf{Set}$.

Corollary 3.8. *For any set* X, $(\mathrm{Gr} \circ \mathcal{H})(X) \cong \mathcal{G}(X)$.

Proof. Since both functors $\mathrm{Gr} \circ \mathcal{H}$ and $\mathcal{G}$ are left adjoints to the forgetful functor, they are naturally isomorphic, see [8, Corollary 1, page 85]. □

4. Conclusion

We have shown that a free functor from the category of sets to the category of groups is decomposable through the category of heaps. That gives us a different view on adding identity and inverses in the free group construction. In the composition of functors $\mathrm{Gr} \circ \mathcal{H}$, first, we take a set X, generate a free heap $\mathcal{H}(X)$, and then consider a coproduct of the free heap with a singleton heap $\mathcal{H}(X) \boxplus \{*\}$. The singleton heap is a unique free heap over a singleton set, that is $\{*\} = \mathcal{H}(\{*\})$. Thus the whole process of taking a free group is the same as taking a free heap over $X \sqcup \{*\}$, as the free heap functor preserves colimits. The inverses of group operation arise naturally as words of the form $w^{-1} = [*, w, *]$ in $\mathcal{H}(X) \boxplus \{*\}$. When constructing the free group, we need to take the following coproducts of sets $X \sqcup X^{-1} \sqcup \{*\}$ and go through the whole process of spanning the word algebra.

The classical construction of a free group is more complicated as we have a bigger set of generators of a word algebra.

Acknowledgements. The author is grateful to the anonymous reviewer, Tomasz Brzeziński and Paolo Saracco, for all the comments and advice. I would also like to thank the editor and anonymous reviewer for their helpful comments.

References

[1] R. Baer, *Zur einführung des scharbegriffs*, J. Reine Angew. Math., 160 (1929), 199-207.

[2] M. Barr and C. Wells, *Toposes, triples and theories*, Corrected reprint of the 1985 original Repr. Theory Appl. Categ., 12 (2005), 1-288.

[3] G. M. Bergman, An Invitation to General Algebra and Universal Constructions, Springer, Cham, Second Edition, 2015, available online, https://math.berkeley.edu/ gbergman/245/3.2.pdf.

[4] T. Brzeziński, *Trusses: paragons, ideals and modules*, J. Pure Appl. Algebra, 224(6) (2020), 106258 (39 pp).

[5] T. Brzeziński and B. Rybołowicz, *Modules over trusses vs modules over rings: direct sums and free modules,* Algebr. Represent. Theory, 25(1) (2022), 1-23.

[6] D. S. Dummit and R. M Foote, Abstract Algebra, Prentice Hall, 1991.

[7] W. Dyck, *Gruppentheoretische studien*, Math. Ann., 20(1) (1882), 1-44.

[8] S. Mac Lane, Categories for the Working Mathematician, Springer, New York, Second Edition, 1998.

[9] H. Prüfer, *Theorie der abelschen gruppen*, Math. Z., 20(1) (1924), 165-187.

Bernard Rybołowicz
Department of Mathematics
Maxwell Institute
Heriot-Watt University
EH14 4AS Edinburgh, UK
e-mail: B.Rybolowicz@hw.ac.uk