

# On commutativity, total orders, and sorting

Wind Wong<sup>1</sup>, Vikraman Choudhury<sup>\*2</sup>, and Simon J. Gay<sup>3</sup>

<sup>1,3</sup>University of Glasgow

<sup>2</sup>Università di Bologna and OLAS Team, INRIA

February 25, 2024

In this talk, we study free monoids, free commutative monoids, and their connections with sorting and well-orders. Univalent type theory provides a rigorous framework for implementing these ideas, in the construction of free algebras using higher inductive types and quotients, and reasoning up to equivalence using categorical universal properties. The main contributions are a new framework for universal algebra (free algebras and their universal properties), various constructions of free monoids and free commutative monoids (with proofs of their universal properties), applications to proving combinatorial properties of these constructions, and finally an axiomatic understanding of sorting. Our results have been formalized in Cubical Agda, and the formalization is available at: <https://github.com/pufferfish/agda-symmetries/>.

## Background

First, we review the basics of universal algebra, free algebras and their universal property. This is standard in the literature on (first-order) universal algebra and equational logic (predating abstract clones and Lawvere theories). We write  $\text{Set}$  for the category of  $\mathbf{hSets}$  and functions. A signature  $\sigma$  is given by a set of operations with an arity function:  $(\text{op} : \text{Set}) \times (\text{ar} : \text{op} \rightarrow \text{Set})$ . This gives a signature endofunctor  $F_\sigma(X) \equiv (f : \text{op}) \times (X^{\text{ar}(f)})$  on  $\text{Set}$ . A  $\sigma$ -structure  $\mathfrak{X}$  (with carrier set  $X$ ) is an  $F_\sigma$ -algebra:  $(X : \text{Set}) \times (\alpha_X : F_\sigma(X) \rightarrow X)$ , and a homomorphism of  $\sigma$ -structures is a  $F_\sigma$ -algebra morphism, giving the category of  $F_\sigma$ -algebras  $F_\sigma\text{-Alg}$ . A concrete example for the case of monoids is given in section 2.

The forgetful functor  $U_\sigma : F_\sigma\text{-Alg} \rightarrow \text{Set}$  (which gives the underlying carrier set), admits a left adjoint, that produces the free  $\sigma$ -structure (or  $F_\sigma$ -algebra) construction on a carrier set. As is standard, this construction is given by an inductive type of trees  $\text{Tr}_\sigma(V)$ , generated by two constructors:  $\bullet$  leaf:  $V \rightarrow \text{Tr}_\sigma(V)$ , and  $\bullet$  node:  $F_\sigma(\text{Tr}_\sigma(V)) \rightarrow \text{Tr}_\sigma(V)$ .  $\text{Tr}_\sigma(V)$  is canonically an  $F_\sigma$ -algebra:  $\mathfrak{T}(V) = (\text{Tr}_\sigma(V), \text{node})$ , free on the set of generators  $V$ , with the universal map  $\eta_V : V \rightarrow \text{Tr}_\sigma(V)$  given by leaf. The universal property states that, given any  $\sigma$ -structure  $\mathfrak{X}$ , composition with  $\eta_V$  is an equivalence:  $(-) \circ \eta_V : F_\sigma\text{-Alg}(\mathfrak{T}(V), \mathfrak{X}) \xrightarrow{\sim} (V \rightarrow X)$ . The inverse to this map is the extension operation  $(-)^{\sharp}$ , which extends a map  $f : V \rightarrow X$  to a homomorphism  $f^{\sharp} : \mathfrak{T}(V) \rightarrow \mathfrak{X}$ .

An equational signature  $\varepsilon$  is given by a set of equations with an arity of free variables for each equation:  $(\text{eq} : \text{Set}) \times (\text{fv} : \text{eq} \rightarrow \text{Set})$ . A system of equations over  $(\sigma, \varepsilon)$  (or a theory  $T_{(\sigma, \varepsilon)}$ ) is given by a pair of trees on the set of free variables, for each equation:  $\ell, \nu : (e : \text{eq}) \rightarrow \text{Tr}_\sigma(\text{fv}(e))$ . A  $\sigma$ -structure  $\mathfrak{X}$  satisfies  $T$ , written  $\mathfrak{X} \vDash T$ , if, for each equation  $e : \text{eq}$  and  $\rho : \text{fv}(e) \rightarrow X$ , we have  $\rho^{\sharp}(\ell(e)) = \rho^{\sharp}(\nu(e))$ . The full subcategory of  $F_\sigma\text{-Alg}$  given by  $\sigma$ -structures satisfying  $T_{(\sigma, \varepsilon)}$  is the variety of  $T_{(\sigma, \varepsilon)}$ -algebras. The free object in the variety of  $T_{(\sigma, \varepsilon)}$ -algebras is, similarly, given by the left adjoint to the forgetful functor to  $\text{Set}$ . Classically, this is constructed by quotienting the free  $F_\sigma$ -algebra by the congruence relation generated by the equations, which requires non-constructive principles (such as assuming the sets of arities support choice, see [1, § 7]). In this talk, we only consider the construction of free algebras for the specific varieties of monoids and commutative monoids, in section 2.

## Monoids and Commutativity

The signature for monoids  $\sigma_{\text{Mon}}$  is given by two operations (unit and multiplication) of arity 0 and 2, respectively, written as  $(\text{Fin}_2, \{0 \mapsto \text{Fin}_0; 1 \mapsto \text{Fin}_2\})$ . The equational signature for monoids  $\varepsilon_{\text{Mon}}$  is given by three equations

<sup>\*</sup>Supported by EU Marie Skłodowska-Curie fellowship 101106046 ReGraDe-CS.

(unitl, untr, assoc) which use 1, 1, and 3 free variables, respectively, written as  $(\text{MonSig}, \{\text{unitl} \mapsto \text{Fin}_1; \text{untr} \mapsto \text{Fin}_1; \text{assoc} \mapsto \text{Fin}_3\})$ . We write  $T_{\text{Mon}}$  for the theory of monoids, which is given by the pairs of left and right trees, using the free variables for each equation. Commutative monoids are given by the same signature of operations, but additionally include the commutativity equation, which uses 2 free variables, written as  $T_{\text{CMon}}$ .

In this framework, we study various constructions of free monoids and free commutative monoids, using HITs and quotients, and prove the universal property for each construction. We construct:

- FreeMon and FreeCMon HITs, given by generators for operations and higher generators for equations, which is the standard construction of free algebras in HoTT [7], and
- List, SList, CList, given by cons-lists, cons-lists with adjacent swaps, cons-lists with a commutation relation, respectively, from [2, 5].

Using the construction of free algebras as quotients, we consider various commutativity relations on various presentations of free monoids. Given a free monoid construction:  $A \xrightarrow{\eta} \mathcal{L}(A)$ , a commutativity (or permutation) relation is a binary relation  $\approx$  on  $\mathcal{L}(A)$  such that,  $A \xrightarrow{\eta} \mathcal{L}(A) \xrightarrow{q} \mathcal{L}(A)/\approx$  is a free commutative monoid construction. From this we construct:

- PList, a quotient of List by various permutation relations, considered in [3, 5], and
- Bag, a quotient of  $\text{Array}(A) = (n : \mathbb{N}) \times (f : A^{\text{Fin}(n)})$  by  $(n, f) \sim (m, g) \equiv (\sigma : \text{Fin}(n) \simeq \text{Fin}(m)) \times (f = g \circ \sigma)$ , considered in [2, 5].

We use these constructions to study various properties of free monoids and free commutative monoids:

- definitions of length,  $\in$ , Any, All, by homomorphic extension,
- characterizations of the path spaces of each type,
- combinatorial properties, such as  $\mathcal{M}(A + B) \simeq \mathcal{M}(A) \times \mathcal{M}(B)$  (dual of Fox's theorem [4]),
- injectivity of  $\text{cons}_A(x, -)$  for any  $x : A$ .

## Total orders and Sorting

Finally, our main result is to use this framework to study sorting and total orders. It is commonly understood that lists are ordered lists and bags are unordered lists. Our aim is to give a conceptual explanation of this fact – commutativity forgets the canonical ordering on an ordered list, and the only way to recover an ordered list from an unordered list is to sort it, which requires a total ordering on the elements.

Given a total order on a set  $A$ , a sorting algorithm, informally, turns lists of  $A$  into sorted lists of  $A$ . Formally, we observe that this produces a section to the canonical homomorphism from the free monoid to the free commutative monoid:  $q : \mathcal{L}(X) \twoheadrightarrow \mathcal{M}(X)$  (see [5]). Turning it around, we ask what it means for this section to be well-behaved – a well-behaved section should recover the total order on the carrier set, which axiomatizes the correctness of a sorting algorithm. Our final theorem is summarized below.

**Definition 1.** *Given a section  $s : \mathcal{M}(X) \rightarrow \mathcal{L}(X)$  to  $q$ , we say that a list  $xs : \mathcal{L}(X)$  is sorted (with respect to  $s$ ), if there merely exists a  $ys : \mathcal{M}(X)$  such that  $s(ys) = xs$ .*

A well-behaved section  $s$  should satisfy the following two axioms, which axiomatizes a correct sorting algorithm.

**Definition 2.**

- is-head-least: *the head of a sorted list is the least element:  $\forall x y xs. \text{is-sorted}_s(x :: xs) \wedge y \in (x :: xs) \rightarrow \text{is-sorted}_s(x :: y :: [])$ ,*
- is-tail-sort: *the tail of a sorted list is sorted:  $\forall x xs. \text{is-sorted}_s(x :: xs) \rightarrow \text{is-sorted}_s(xs)$ .*

Finally, we state our main theorem.

**Theorem 1.** *Let  $\text{DecTotOrd}(A)$  be the set of decidable total orders on  $A$ , and  $\text{Sort}(A)$  be the set of correct sorting functions with carrier set  $A$ . The function  $\text{o2s} : \text{DecTotOrd}(A) \rightarrow \text{Sort}(A) \times \text{isDiscrete}(A)$  is an equivalence.*

To conclude, we will mention some work in progress, on generalizing this framework from sets to groupoids, using a system of coherences on top of the system of equations. As an instance of this, we consider the construction of free monoidal and free symmetric monoidal groupoids.

## References

- [1] Andreas Blass. “Words, free algebras, and coequalizers”. eng. In: *Fundamenta Mathematicae* 117.2 (1983), pp. 117–160. URL: <http://eudml.org/doc/211359>.
- [2] Vikraman Choudhury and Marcelo Fiore. “Free Commutative Monoids in Homotopy Type Theory”. In: *Electronic Notes in Theoretical Informatics and Computer Science* Volume 1-Proceedings of... (Feb. 2023). ISSN: 2969-2431. DOI: 10.46298/entics.10492.
- [3] Nils Anders Danielsson. “Bag Equivalence via a Proof-Relevant Membership Relation”. In: *Interactive Theorem Proving*. Ed. by Lennart Beringer and Amy Felty. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 149–165. ISBN: 978-3-642-32347-8.
- [4] Thomas Fox. “Coalgebras and cartesian categories”. In: *Communications in Algebra* 4.7 (1976), pp. 665–667. DOI: 10.1080/00927877608822127. eprint: <https://doi.org/10.1080/00927877608822127>. URL: <https://doi.org/10.1080/00927877608822127>.
- [5] Philipp Joram and Niccolò Veltri. “Constructive Final Semantics of Finite Bags”. In: *14th International Conference on Interactive Theorem Proving (ITP 2023)*. Ed. by Adam Naumowicz and René Thiemann. Vol. 268. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023, 20:1–20:19. ISBN: 978-3-95977-284-6. DOI: 10.4230/LIPIcs.ITP.2023.20. URL: <https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ITP.2023.20>.
- [6] Clemens Kupke, Fredrik Nordvall Forsberg, and Sean Watters. “A Fresh Look at Commutativity: Free Algebraic Structures via Fresh Lists”. In: *Programming Languages and Systems: 21st Asian Symposium, APLAS 2023, Taipei, Taiwan, November 26–29, 2023, Proceedings*. Taipei, Taiwan: Springer-Verlag, 2023, pp. 135–154. ISBN: 978-981-99-8310-0. DOI: 10.1007/978-981-99-8311-7\_7.
- [7] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. Princeton, NJ: Institute for Advanced Study, 2013. ISBN: 978-0-8176-4360-7. URL: <https://homotopytypetheory.org/book/>.