Stream Ring Theory, Correctly

Marko and Ryan

May 30, 2020

Abstract

In this paper we correct the mathematical errors in [4], resulting in a theory similar in spirit. All our proofs are formalized in the Coq proof assistant [2]. Familiarity with category theory at the level of [1] is required to understand this paper. This paper is a work in progress; we have only formalized the first few sections of [4].

1 Bags and Streams

Let X be a set. A *bag* [3] (or *multi-set*) over X is a possibly empty list (ordered sequence) $[x_1, \ldots, x_n]$, with every x in X, considered up to permutation. In this paper, we refer to the underlying list as the bag's *stream*, to each x as a *stream object*, to the set of all bags over X as $\mathsf{Bag}(X)$, and for every set X, to every function of the form $f : Y \to \mathsf{Bag}(X)$ as a *stream function* from Y to X.

As examples, the stream function $\mathsf{cons} : X \times \mathsf{Bag}(X) \to \mathsf{Bag}(X)$ inserts a stream object into a bag, and the stream function $\mathsf{remove} : X \times \mathsf{Bag}(X) \to \mathsf{Bag}(X)$ removes an object from a bag, if present, otherwise returning its input.

As a technical convenience, and without loss of generality, in this paper we will assume that every set X is equipped with a total order $\preceq_X$, and that every bag's stream is sorted with respect to $\preceq_X$. This assumption allows us to define a unique representative list for each bag, facilitating the Coq proofs, and follows from the axiom of choice and classical logic.

Lemma 1. The stream functions form a category $\mathcal{F}$ (depending on $\preceq$) as follows:

- The objects of $\mathcal{F}$ are all the sets.
- The arrows of $\mathcal{F}$ from X to Y are all the stream functions from X to Y.
- Composition of $f : X \to \mathsf{Bag}(Y)$ and $g : Y \to \mathsf{Bag}(Z)$ given by:

$$(f;g)(x \in X) \in \mathsf{Bag}(Z) \ := \biguplus_{y \in f(x)} g(y),$$

  where $\uplus$ indicates bag union (appending and sorting the underlying streams).
- The identity $\mathsf{id}_X : X \to \mathsf{Bag}(X)$ is the function taking each $x \in X$ to $[x]$. □

The next three lemmas prove that $\mathcal{F}$ has all finite products and co-products.

Lemma 2. The empty set is initial and final in $\mathcal{F}$. □

Lemma 3. For all sets X and Y, the disjoint union $X + Y$ and stream functions:

$$\iota_1(x \in X) \in \mathsf{Bag}(X+Y) := [\mathsf{inl}(x)] \quad \iota_2(y \in Y) \in \mathsf{Bag}(X+Y) := [\mathsf{inr}(y)]$$

form a coproduct in $\mathcal{F}$, where for every set Z and stream functions:

$$i_1 : X \to \mathsf{Bag}(X+Y) \quad i_2 : Y \to \mathsf{Bag}(X+Y)$$

the case analysis morphism $X + Y \to Z$ sends each $x \in X$ to $i_1(x)$ and each $y \in Y$ to $i_2(y)$. □

Lemma 4. For all sets X and Y, the disjoint union $X + Y$ and stream functions:

$$\pi_1(x \in X{+}Y) \in \mathsf{Bag}(X) := \mathsf{if}\ x \in X\ \mathsf{then}\ [x]\ \mathsf{else}\ [] \quad \pi_2(x \in X{+}Y) \in \mathsf{Bag}(Y) := \mathsf{if}\ y \in Y\ \mathsf{then}\ [y]\ \mathsf{else}\ []$$

form a product in $\mathcal{F}$, where for every set Z and stream functions:

$$p_1 : Z \to \mathsf{Bag}(X) \quad p_2 : Z \to \mathsf{Bag}(Y)$$

the pairing morphism $Z \to X + Y$ sends each $z \in Z$ to $\mathsf{bmap}(\mathsf{inl}, p_1(z)) \uplus \mathsf{bmap}(\mathsf{inr}, p_2(z))$, where

$$\mathsf{bmap}(f, [v_1, \ldots, v_n]) := [f(v_1)] \uplus \ldots \uplus [f(v_n)]$$

denotes the bag-mapping function. $\square$

In this paper, we are primarily interested in the fact that $\mathcal{F}$ is a *ringoid*: a category where each hom-set is an abelian group:

Lemma 4. For all sets X and Y, the functions $X \to \mathsf{Bag}(Y)$ form an abelian group, with $(f + g)(x) := f(x) \uplus g(x)$ and $0(x) := []$. $\square$

References

[1] Steve Awodey. *Category Theory*. Oxford University Press, Inc., USA, 2nd edition, 2010.

[2] Yves Bertot and Pierre Castran. *Interactive Theorem Proving and Program Development: Coq'Art The Calculus of Inductive Constructions*. Springer Publishing Company, Inc., 2010.

[3] Leonid Libkin and Limsoon Wong. Query languages for bags and aggregate functions. *Journal of Computer and System Sciences*, 55:241–272, 04 2000.

[4] Marko A. Rodriguez. Stream ring theory, `https://doi.org/10.5281/zenodo.2565243` 2019.

2 Appendix: Coq Assumptions

```
zorn : forall T : Type, exists R : relation T, well_order T R

proof_irrelevance : forall (P : Prop) (p1 p2 : P), p1 = p2

functional_extensionality_dep : forall (A : Type) (B : A -> Type)
   (f g : forall x : A, B x),  (forall x : A, f x = g x) -> f = g

em : forall P : Prop, {P} + {~ P}

constructive_indefinite_description : forall (A : Type) (P : A -> Prop),
   (exists x : A, P x) -> {x : A | P x}
```