A Concise Depiction of Cosmological Experiments and Associated Theories

Ben Sprott

benjamin.sprott@mail.mcgill.ca

November 11, 2018

Abstract

Familiar data structures from computer science are cast in the form of Monads and Comonads and their application to laboratory physics is demonstrated. These data structures are then used to model a cosmological thought experiment introduced by Hardy, encoding and relating fixed causal structure and structure free situations. The data structures are harmonized in a diagrammatic calculus and the thought experiment is calculated therein.

1 Introduction

It has been shown that experiments naturally occur from the evolution of a discrete causal background [15]. Since experiments are equivalent to endofunctors that are both monads and comonads on the category of the apparatus one uses, information naturally arises. The co-product of the comonad represents copying information and the product represents the coalescing of different instances of the experiment into a monolith of data.

Quantum Theory is a special type of probability theory. As such, information is deeply embedded in physics. Because of this, there has been much work on prizing apart exactly what aspects of physics can be said to be informational and which parts are physical. This has lead to investigations of the reality of the wavefunction [12] [11], and informational derivations of quantum theory [2]. This paper extends work on the idea that information is embedded in physics in a natural way if we view physics as concerning causality and the fact that observers have limited access to the information concerning causal relationships.

2 Data Structures and (Co)Monads

Data structures come in many different flavours. The prototypical data structure is the List. It is well known that Lists can be encoded as Monads and we call it the List Monad. Work has been done to encode certain data structures, called containers as Monads [5]. There is also a prescription for when a data structure is a Comonad [1].

2.1 Examples of Data Structures

There are a few data structures we are going to use in this paper. One such structure, that captures the idea of a fixed, universal clock is the List data structure and we will discuss it below. Another such data structure is meant to model the situation where we have data, but no ordering (think causal) relationship between the data. This structure is known as a Bag, see [13]. In that definition the data structure has only an operation to add an element and then an operation to return an element from the bag with no guarantee as the order of how the elements are returned. The Bag data structure can be captured in the Free Commutative Monoid Monad.

2.1.1 Bags

We have a functor $Bag : Set \to Set$, that maps the category Set to itself. It works by taking a set and returning a set of Bags for that set. A bag can be modeled as a commutative monoid, where we have a string of set elements, whose order doesn't matter, so $abc = bca$. The multiplication for a Bag is understood by thinking of free commutative monoids. The first application of the functor $Bag : Set \to Set$ maps a set of elements to the set of formal sums of the elements. For instance, if $A \in Set$, and $A = \{a\}$ then $Bag(A) = \{a, 2a, 3a \cdots\}$. Two applications gives formal sums of formal sums, so $Bag \cdot Bag(A) = \{((a) + (a)), ((a) + (2a))\}$. The multiplication just adds everything together to get back $\{a, 2a, 3a \cdots\}$, so $\mu : Bag \cdot Bag \to Bag$. The unit is the inclusion of the free generators. For a better description see [16].

2.1.2 Lists

Lists are ordered series of objects or symbols. Here is an example of a list on the set $A = \{a, b\}$:

$$[a, b, b, b, a]$$

Lists can be cast as monads. Here is the definition of the List monad.

Definition 2.1. The List Monad is a triple (L, μ, η) where:

$$L : Set \to Set$$

example

$$[a, b, b, a, a, a] \in L(\{a, b\})$$

$$\mu : L \cdot L \to L$$

example

$$[[b, b, a], [a, a]] \in L \cdot L(\{a, b\})$$

$$\eta : 1_{Set} \to L$$

$$\eta : \{a, b\} \to \{[a], [b]\}$$

2.2 Histories as a Comonad

We can convert Lists into Histories with a comonad. For any list l, there is a list of lists of initial partial fragments. For example, let $\mu_H : List \to List \cdot List$ be a natural transformation that takes the $List$ functor to its composition with itself.

$$\mu_H : [a, b, c] \to [[a], [a, b]]$$

This converts the List monad into a bimonad (both a monad and comonad), where μ_H is the coproduct natural transformation.

The counit is the following:

$$\eta_H : [l_1, l_2, l_3] \to l_3$$

ie, it picks out the last element of each list. For this all to work we use non-empty lists, as we see here [8].

2.3 Instruments and Readouts

When we approach an instrument in a laboratory we see that there are a few different aspects to it. The most obvious is the control and readout panel on the front. This panel has a well defined configuration at any point during an experiment. Furthermore, we understand "local", in "local lab", to mean the existence of a universal clock or reference frame. Thus, a local experiment is a totally ordered list of configurations of the equipment. During an experiment the panel goes through a number of changes, and the readout flashes from state to state. Thus, we model this as a list, and further with the List Monad. There is a list of configurations that occur during an experiment. We record each of these configurations. We accept that local labs have a fixed reference frame and a local clock. Thus, it makes sense to use a list for local data. Though there seems to be a fixed order to the set of configurations and readouts, when this data "escapes" the front panel, for instance as we write down the configuration, the ordering of events is lost. We could record the output of a Geiger counter on a piece of paper, as a list of counts with commas between the integers, this ordering is artificial. Thus, after the data has escaped the instrument, it is more like a bag data structure. There is a map from the List monad to the Bag monad that captures this.

2.4 Domains and SpaceTimes

A domain, also known as a dcpo is defined as follows:

Definition 2.2. DCPO (Domain) A *directed subset* of a partially ordered set is a nonempty subset which contains an upper bound for every pair of elements in it. A *domain* or *dcpo* (*directed-complete partial order*) is a partially ordered set such that every directed subset has a supremum.

Panangaden and Martin showed, [10], that the category of hyperbolic space-times is equivalent to the category of Interval Domains. It may not be clear how to interpret this. Though it may not be easy to show weather or not we can do this, we take a radical departure from this work to state that a toy relativistic field theory is any domain of structured objects. We are going to take Lists and Bags to be the basic data structures we work with in science. We consider two laboratories, separated by distance, to be elements of a domain of Lists and elements of a domain of Bags. They are embedded in the Universe, and this is captured by the fact that they are elements of a domain.

2.4.1 A Domain of Lists

There is a natural ordering of Lists via partial initial fragment. Given a set of lists S_L, and two lists $l_1, l_2 \in S_L$, we define the ordering relation:

$$l_1 \leq l_2 \text{ if } l_2 = Concat(l_1, l_2^{'})$$

Concat is the concatenation function for lists.

This forms a DCPO.

2.5 Diagrammatic Calculus

We can consider all the data structures, known as containers, that can be encoded as either monads or comonads or both. These are all defined with an endofunctor on the category of Sets. We define the category of such containers as CON. This category is monoidal where the monoidal product is actually functor composition. It is a regular monoidal category (i.e. not symmetric). Because of this, there is a diagrammatic calculus for this category, that of a regular monoidal category. For this reason, we will now write composition of monads with the product symbol familiar to those working with symmetric monoidal categories, namely $\otimes$. For instance, the product of two functors can be written $Dom \otimes Dom$.

2.6 From Instrument Events to Data

Events in our laboratory all happen in a local frame with a universal clock for all events. This means that there is a list of configurations of the instruments. Thus, histories of events is really the non-empty list bimonad. However, there is no ordering on the data that is produced and collected. Thus we need a map from the List Monad to the Multiset or Bag monad. This is just a natural transformation from the List functor to the Bag functor. It acts just by forgetting the ordering of the list.

2.7 From Data to Probabilities

Typically, when we finish our experiment, we analyze the data from the instruments. This means applying probability theory. Luckily, probability theory fits in quite nicely with structures we are presenting here. The most famous categorical structure that captures probabilities is the Giry Monad. It's core is a functor that maps a set to the set of probability measures on that set. The product axiom can be interpreted in the following way [14]:

> A probability measure on an affine space has an average value (also called expectation or integral), which is a point in that affine space. Apply this to the affine space of probability measures on X. In other words, a measure on the space of measures determines a (weighted) average of those measures.

The final form of the data in our model is that of the Bag data structure or Multiset. We might want to define a transformation from the Bag monad to the Giry monad. Every Bag has an associated probability measure that gives the probability of finding one of the set elements in that Bag. This might imply that there is a transformation from the Bag monad to the Giry monad. However, that is not possible. In it's place, one can use a certain kind of Measure monad as explained in [7]. Thus, there is a map from the Bag Monad to a certain Measures Monad. It is a sensible choice to impose the fact that the collection of cards in any experiment is finite. For this reason, we are interested in finite Bags and finite Lists. This prevents us from talking about a domain of Lists. Of immediate concern is the map from data to probabilities, so we take this as the natural transformation from the finite Bag functor to the Measures of Finite support functor. We call this natural transformation *DataToProb*. We will call the Finite Measure Monad $FinMeas$.

3 A Cosmological Thought Experiment

3.1 The Experiment

In his paper on probability theories without fixed causal structure [6], Hardy introduces a kind of cosmological thought experiment. The interesting thing about this thought experiment is its intimate relationship to data and the actions of a group of scientists working together. In the thought experiment, we are asked to imagine several scientists, or perhaps just laboratories, scattered around the galaxy. Causal relationship between these scientists is unknown, but we assume that we can communicate. The items we wish to communicate are the results of local experiments. These experiments might involve looking at the same thing, like say a black hole at the center of the galaxy. Thus, we take down recordings of our local lab equipment. These recordings, or cards, are shuffled locally and then gathered into a single pile and shuffled again. How can we capture all of this?

Since hyperbolic space-times are equivalent to interval domains, we can imagine a domain that connects all the local laboratories. We know there is a domain of Lists, and this models Hardy's picture of a number of labs with instruments collecting data on cards. The card can be the element of the list. The shuffling of the lists is a map from List to Bag, which is forgetting the ordering. Thus, we know that each lab is really a *Bag* of readings on equipment. Collecting the cards together is a map from Domain to Bag, which forgets the causal relations between the labs and collects the card packs into a jumble of cards. Thus, we have different monads that capture different aspects of the cosmological experiment. The collecting of cards will be handled by monad maps and the product axiom of the bag monad.

We can organize all of these Monads together in the string calculus for the category of Containers. We have data types *Bag*, *Dom*, *List* which are the monads. We also have monad transformations defined as follows.

$$DomToBag : Dom \to Bag$$

This forgets the domain structure and returns just a bag of whatever was contained in the domain.

$$ListToBag : List \to Bag$$

This takes a list and returns the bag for that list.

$$\mu : Bag \otimes Bag \to Bag$$

This is just the multiplication of the Bag monad. It takes a bag of bags and dissolves the inner bags and returns one large bag.

The thought experiment then looks like the string diagram found in figure 1.

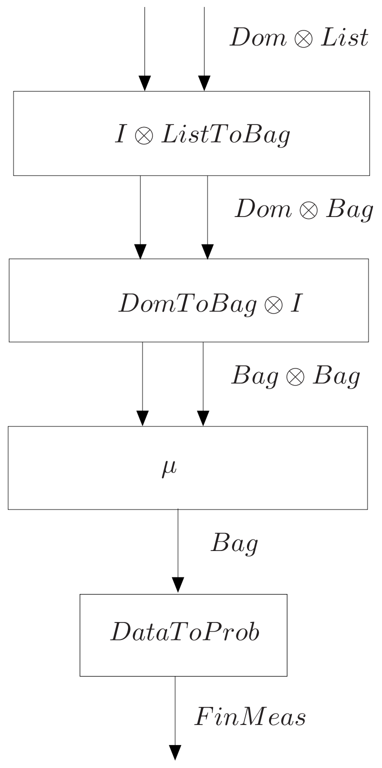

Figure 1: A Cosmological Thought Experiment

Thus, we start with a Domain of Lists which represents the set of causally connected labs with a totally ordered set of instrument configurations. This gets mapped to a Domain of bags, as the total order of configurations gets turned into an unordered collection of cards, modeled as a bag of cards. Then we collect all the cards from all the laboratories and this is forgetting the causal relations between all the laboratories and thus we map Dom to Bag to have a bag of bags. Then we use the monad multiplication to disolve all the inner bags, thus shuffling them, and we get on large unordered collection of cards, namely a bag of cards. Finally, we want a probability distribution for this collection of cards and this is done just by using the monad map that maps the Bag monad to the Measures Monad to give the final string in the diagram $FinMeas$.

3.2 Theories, Both Mechanical And Probabilistic

Everything depicted in figire 1 pertains to local information and actions. What is entirely missing is a discussion about the distant, imprecisely understood system that is being observed. The disparate experiments could be looking at a black hole at the center of the galaxy, or they could be looking at the earliest possible events in the universe. When we think of a distant, imprecisely known system, to which we have limited access, we have to consider two viewpoints simultaneously. We have to consider that our reasoning about this distant system is via a theory only. Furthermore, we accept that there objectively exists processes in the universe that we can observe, that are the real kernels for this theory. During an experiment, we access those objective processes through our local instruments and the events surrounding those instruments.

One of the very bold assertions in [15], is that, once you have the general prescription for your experiment in terms of a (co)monad, the associated theory is essentially a calculation away. This is because every monad is the result of some adjunction. A term for this effect is the "factorization" of the monad. The result of the factorization of a monad M, on a category $\mathcal{C}$ is a category $\mathcal{D}$ and an adjunction F, G, where $F : C \to D$, $G : D \to C$, such that $G \cdot F = M$. For a finite category C, there is even open source software that will calculate this factorization [3].

Let's look at two of the stages in the calculation shown in figure 1, namely the single Bag string and the single $FinMeas$ string.

3.2.1 "Doing" Finite Measure Probability Theory

The monad that is associated with the probabilities we expect to calculate in our experiment is the "Measures of Finite Support Monad". What is the probability theory for this monad? We expect that we can calculate this theory by a factorization of the monad. In particular, we expect that the most suitable Category of Algebras for this monad is the Eilenberg-Moore category for $FinMeas$. This category is known and is analyzed in [4] in terms of the $\mathcal{G}_{fin}$-algebra.

3.2.2 The Mechanical Theory

In this work, we take the position that the physical aspect to the theory comes before the probabilistic aspect. That is to say, we have a monad of data that is associated with our apparatus and it gives rise to a probability monad via a natural transformation. Thus, the mechanical aspect to the theory is encoded in the monad that is upstream to the probability monad. This means that we are going to look for the category of algebras for the finite Bag monad. This monad is the finite multiset monad, or free commutative monoid monad. Furthermore, along with this monad comes a notion of multiplication by the semi-ring of natural numbers and also addition. The idea being that the multiset $m = \{a, a, b, b, c\}$ on the set $S = \{a, b, c\}$, can equally be given as a formal sum $m = 2a + 2b + c$. Because of this, we see that the category of algebras for the multiset monad is is th category of $\mathbb{N}$-modules, as suggested in [9].

It is hard to say what physical significance there is in the category of $\mathbb{N}$-modules. Modules are something like vector spaces, which appear frequently in physics. There is some evidence that the particle theory of physics is encapsulated in this category of algebras. For instance, supposing you had many different types of particle, you might want to keep track of a collection of them. To that end you may group them. Let's label the types of particles with colours, then a collection of particles is a formal sum like this $c = \{blue, blue, red, green, green, green\} = (2blue + red + 3green)$. What is good news is that the category of $\mathbb{N}$-modules is fairly richly structured, being a module and having the natural numbers. We could probably model a fair bit of physics this way. It can be noted that the category of $\mathbb{N}$-modules is equivalent to the category of abelian groups.

References

[1] Danel Ahman, James Chapman, and Tarmo Uustalu. When is a container a comonad? In *International Conference on Foundations of Software Science and Computational Structures*, pages 74–88. Springer, 2012.

[2] G. Chiribella, G. D'Ariano, and P. Perinotti. Informational derivation of quantum theory. *Physical Review A*, 2011.

[3] Ryan Wisnesky David Spivak. Categoricaldata fql.

[4] Tobias Fritz. Convex spaces i: Definition and examples. *arXiv preprint arXiv:0903.5522*, 2009.

[5] Nicola Gambino and Joachim Kock. Polynomial functors and polynomial monads. In *Mathematical proceedings of the cambridge philosophical society*, volume 154, pages 153–192. Cambridge University Press, 2013.

[6] Lucien Hardy. Probability theories with dynamic causal structure: a new framework for quantum gravity. *arXiv preprint gr-qc/0509120*, 2005.

[7] Peter LeFanu Lumsdaine (https://mathoverflow.net/users/2273/peter-lefanu lumsdaine). Map from the multiset monad to the giry monad: From data to probabilities. MathOverflow. URL:https://mathoverflow.net/q/310888 (version: 2018-09-18).

[8] Tom Leinster (https://mathoverflow.net/users/586/tom leinster). List is a monad, but is it a comonad with these natural transformations? MathOverflow. URL:https://mathoverflow.net/q/237967 (version: 2016-05-03).

[9] Bart Jacobs. Bases as coalgebras. In *International Conference on Algebra and Coalgebra in Computer Science*, pages 237–252. Springer, 2011.

[10] Keye Martin and Prakash Panangaden. A domain of spacetime intervals in general relativity. *Communications in mathematical physics*, 267(3):563–586, 2006.

[11] Matthew Pusey, Jonathan Barret, and Terry Rudolph. On the reality of the quantum state. *Nature Physics*, 2012.

[12] et al. Ringbauer, Martin. Measurements on the reality of the wavefunction. *Nature Physics*, 11.3:249–254, 2015.

[13] Robert Sedgewick. *Algorithms*. Pearson Education India, 1988.

[14] Ben Sprott. https://mathoverflow.net/questions/310704/map-from-the-multiset-monad-to-the-giry-monad-from-data-to-probabilities. *https://math.stackexchange.com/questions/2901834/interpretation-of-the-product-axiom-for-the-giry-monad*, 2018.

[15] Benjamin Sprott. Experiments and theories, a fundamental model. *http://benjaminsprott.com/ExperimentsAndTheoriesA_FoundationMarch18_2017tex.pdf*, 2017.

[16] Eric Wofsey. free commutative monoid monad. *https://math.stackexchange.com/questions/2634630/free-commutative-monoid-monad*, 2018.