COMPUTATIONAL TRINITARIANISM

JACKSON VAN DYKE

This is a highly condensed version of [BS09].

1. Category theory

Definition 1. A *category* $\mathbf{C}$ consists of:

- a set (or class) of *objects* (if X is an object we write $X \in \mathbf{C}$) and
- for every pair of objects $X, Y \in \mathbf{C}$, a set of *morphisms from X to Y*, written $\operatorname{Hom}(X, Y)$. For $f \in \operatorname{Hom}(X, Y)$ we write $f : X \to Y$.

These have to satisfy the following:

- for every objects there is an identity morphism $1_X \in \operatorname{Hom}(X, X)$;
- for $f : X \to Y$ and $g : Y \to Z$, there is a composite morphism $gf : X \to Z$;
- composition is associative: $(hg) f = h (gf)$.

Example 1. There is a category **Set**. The objects are sets, and the morphisms are set-theoretic functions.

Example 2. There is a category $\mathbf{Vect}_k$. The objects are vector spaces over a field k, and morphisms are linear maps.

Example 3. There is a category **Hilb**. The objects are (finite dimensional) Hilbert spaces and the morphisms are linear maps (automatically bounded in this case).

Definition 2. Let $\mathbf{C}$ and $\mathbf{D}$ be categories. A functor $F : \mathbf{C} \to \mathbf{D}$ is a function on the sets of objects such that for any morphism $f : X \to Y$ there is a morphism $F(f) : F(X) \to F(Y)$ such that

- for any $X \in \mathbf{C}$, $F(1_X) = 1_{F(X)}$ and
- for any two morphisms $f : X \to Y$ and $g : Y \to Z$,

$$F(gf) = F(g) F(f) \ . \tag{1}$$

The point is that categories consist of things, and ways to get between things. Physics, logic, and computer science (type theory) also all have things, and processes between them. If we ask for different notions of things and processes in these settings, they will correspond to asking for different structures on the corresponding category.

For example, it is possible to formalize a notion of a categories having a "product structure" (in the case of **Vect** or **Hilb** this is literally the tensor product of vector spaces).

Date: Last edited: September 7, 2020.

2. Physics

Write the dimension of space as $n-1$. Then we can imagine that time contributes a single additional dimension, so spacetime is dimension n. But we don't want to allow any n-dimensional manifold to be spacetime. Instead, we restrict our attention to what is called a *cobordism*.

Definition 3. Let M and N be $n-1$-dimensional manifolds. A *cobordism between M and N* is a manifold with boundary W of dimension n such that

$$\partial W = M \sqcup N \ . \tag{2}$$

Example 4. Let $n = 1$. This means space is dimension 0, i.e. points, and a cobordism is a line between points (or a circle, which is regarded as a cobordism from the empty set to itself).

Example 5. Let $n = 2$. This means space is dimension 1, i.e. lines or circles, and a cobordism is a surface with those circles or lines as its boundary.

Theorem 1. *There is a category* $\mathbf{Cob}_n$*. The objects are $n-1$-dimensional manifolds and the morphisms are cobordisms between them.*

Warning 1. We are leaving our many details about cobordism categories. There are many types of them given by what we ask for from the objects, e.g. a smooth structure, a metric, the conformal class of the metric, etc.

Definition 4. A quantum field theory is a functor

$$Z : \mathbf{Cob}_n \to \mathbf{Hilb} \ . \tag{3}$$

That is, it sends space to a Hilbert space of states, and it sends spacetime to a linear map between Hilbert spaces (of bounded norm).

Example 6. Quantum mechanics is a 1-dimensional QFT in this sense.

Remark 1. This is the first layer of a very fat onion of (∞, n)-categories and the cobordism hypothesis.

3. Logic

Propositional calculus allows one to reason with abstract propositions. Given a starting collection of propositions, we can build propositions out of them using connectives such as $\wedge$, $\vee$, $\implies$, $\neg$, $\top$, and $\bot$. Then there are some formal rules (e.g. Modus Ponens) which tell us how we are allowed to manipulate propositions to prove other propositions.

Given a flavor of logic (i.e. choice of inference rules, etc.), we can build a category. In particular, we take the propositions as objects, and take (equivalence classes of[1]) proofs as the morphisms.

Example 7. Intuitionistic propositional logic (i.e. propositional logic with no law of excluded middle) corresponds to a cartesian closed category.

[1] The equivalence relation is generated by some specified inference rules.

As we add or remove more rules, and indeed pass to more complicated settings such as predicate logic, we get different flavors of categories.

4. Type theory

In certain forms, type theory offers an alternative to set theory as a foundation for mathematics. Very roughly speaking, this goes as follows. We start with an arbitrary set of *types*. We should think of a type as a collection of things sharing a similar property. When a is of type X, we write $a : X$ and say a is a term of type X. We will insist on having a trivial type I. We will also insist on having a term $1 : I$.

Example 8. There is a natural number type, an integer type, a group type, etc. The natural number 2 is a term of type $\mathbb{N}$, i.e. $2 : \mathbb{N}$.

Then we have rules for forming new types from given types. For example, given two types X and Y, we can form the function type $X \to Y$ and the product type $X \otimes Y$.

Given a particular type theory, there is a category associated to it. The types are the objects, and the morphisms between two types X and Y are given by (equivalence classes of[2]) terms of type $X \to Y$. We should think of the morphisms as *programs* which take in data of type X and output data of type Y.

5. Rosetta stone

In summary, we have the following Rosetta Stone[BS09]:

Category th'y	Physics	Topology	Logic	Computation
object	Hilbert space	manifold	proposition	data type
morphism	operator	cobordism	proof	program
$\otimes$	$\otimes$	disjoint union	conjunction	product
terminal obj.	$\{0\}$	$\emptyset$	True	triv. type I
initial obj.	$\{0\}$	$\emptyset$	False	empty type

References

[BS09] John C. Baez and Mike Stay. Physics, Topology, Logic and Computation: A Rosetta Stone. *arXiv e-prints*, page arXiv:0903.0340, Mar 2009. 1, 3

[2]Two terms are considered equivalent if they differ by some given *rewrite rules*.