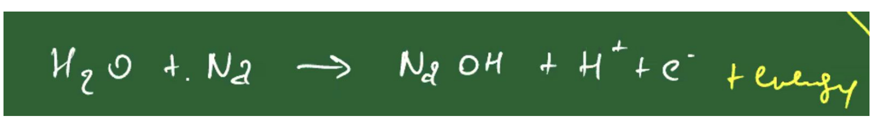
Applied Compositional Thinking for Engineers (ACT4E)

Guest Lecture 3 - Dr. Paolo Perrone

Questions & Answers

AC: BTW I forgot to say Paolo is currently at Oxford with the group of Staton.
Link to Paolo's Category Theory notes: https://arxiv.org/abs/1912.10642

Don't try this at home



Q: This is an off-topic question. Which app is Paolo using to draw and which one was Andrea using in his lectures?

GZ: Andrea and I edit directly on Keynote. Jonathan uses notability.
AC: I think he uses a Windows tablet.
P: Xournal, xournal.sourceforge.net
AR: Why not Xournal++?

Q: are the Kleisli morphisms in C? Do they form a subcategory?
GZ: Yes to the first one. As an example, consider Set and consider the monad Paolo was talking about (using Powerset). Given sets X,Y, a Kleisli morphism would have the form X->PY (assigning to each element of X a subset of Y). This is still a morphism in Set (it is a function).

JL: The Kleisli morphisms with Kleisli composition are not a subcategory of C though. E.g. identity morphisms in C will not in general be a Kleisli morphism.

NM: As a comment to JL’s answer, the identity morphism will only be a Kleisli morphism if and only if TX = X
P: In fact, the original category is a subcategory of the (larger) Kleisli category (generalized morphisms include the old ones)

Q: Why are monads monoids in the category of endofunctors?

NM: The definition of a monoid in a monoidal category C is a triple (M, \mu, \eta) of object M in C, and morphisms \mu: M \otimes M -> M, \eta: 1->M with some requirements on how things compose. The category of endofunctors is a monoidal category where A\otimes B is the endofunctor BA and the identity 1 is the identity functor. The rules for a monad are the same requirements as a monoid where the endofunctor T is the object M.
This might be nice example for tomorrow
Note: another nice example of non-symmetric monoidal categories is 2D topological quantum computing (its braided instead of symmetric)

Q: In the chemical reaction example with the action monad, how can we handle coupling of exergonic and endergonic reactions? I.e. the first reaction releases energy, but the second requires energy. The second reaction will not start spontaneously, without being coupled to the exergonic. In the example shown we can simply add the energies, but now we need to subtract. If R is only a monoid, we can only add. How do we subtract? And how do we express that the endergonic reaction will not happen, unless there is *enough* energy provided?

NM: similar to the money example below combine the monad with the maybe monad. This could be done by allowing R to be negative but define TX = (X, R, possible). Then composition will be as before (allowing for adding -ve numbers) but if there isn’t enough energy for the reaction then set possible to false, and the kleisli morphisms will always set possible to false if it started in false, otherwise it checks if there is enough energy, and if not sets it to false (but still keeps track of the state and the energy).
If you want you could probably also work something in that tracks the true state and doesn’t update if possible is set to false.

How do we express that I *first* need to work to earn money, and only afterwards I can spend money to eat?
Use the “maybe monad”, coupled with the action monad (or modify your monoid)
Outcomes: the usual outcomes or nothing at all

KL: This seems somehow related to the previous lecture with the boolean profunctors of resources and functionality. “Work” provides money and “eat” requires money so they compose nicely if there is enough money to buy food. I wonder if it is always possible to rewrite any monad example to a boolean profunctor though...
NM: I think if you can write your monad as TX = V^Xop for some category V (where X is also a category) then it is a profunctor Xop x X -> V. But maybe this is something to be talked about

elsewhere (and raises the question of where it is useful). This might also mean all profunctors are monads (or maybe 2-monads)
KL: Maybe the concept of lenses is relevant here, since money (or free energy in the chemical example) is regarded as a common resource pool for many processes.

Q: Is there a monad which extends points x,y,z,.. by using graphs or maybe some other combinatorial objects? -> Tree monad: nice!
What is the relation of the tree monad to David Spivaks operads? Here we compose trees by adding to the root?
A reference: Tom Leinster, “higher operads, higher categories” https://arxiv.org/abs/math/0305049
Work of Mark Weber on cartesian monads - these are rather technical texts!

AC: http://eugeniacheng.com/guidebook/ **Higher dimensional categories: an illustrated guide book.**

Q: What are some examples of monads in practice where using the monad construction is better than using the traditional domain approach? (and why?)
OP: domain approach = how the engineer would normally do it (without using monads)

AC: there's a monad of uncertainty intervals. Maps a poset into posets of intervals. (we have seen this in one of the lectures). Defining the monad, then all other collateral concepts are already defined.