Applied Compositional Thinking for Engineers (ACT4E)

Session 0 - Hello!

Questions & Answers

About the contents of the class

Q: Will this course be covering adjoint functor pairs? I understand that they are related to optimization problems.

+4, I also would like to see this

AC: we will find some adjoint functor pairs in the materials. Didn't plan to go in deep but would reconsider.

Q: No Yoneda Lemma? :D

(Vincent Braunack-Mayer): Doesn’t look like it in the syllabus. But it’s only a “trivial exercise” (tongue-in-cheek, given how many cat theory texts are written) :(

+2 to this too!

AC: We definitely will do it, if we find a cool application in engineering. Please tell me if you know one.

Q: Will Operads be a topic in this course? Think boxing components into subsystems? (+1)

AC: It's a bit tight. It would certainly be included if we had one week more.

Optional extra week? :)

AC: maybe :-)

Q. Do you guys link this lecture to a language like sysML or a method like Capella

AC: It seems to me that a comparison of different modeling languages could be in topic with the class. Probably the class would stick to the math due to time bounds, but we would create some extra materials about formal languages.

Q. Will we study Kan extensions and their relation to optimization?

AC: If you know what Kan extensions are, you are likely to recognize them in the materials even if we don't explain them.

Q: Will there be any functional programming connections? I've heard a lot of ties to Category Theory, but I've never used any language. I would love to see the connection.

AC: probably not, because there is a lot of materials about functional programming and CT already. I would suggest the blog (now in book form) of Bartosz Milewski as well as the Youtube videos. See course webpage for links.

Technical questions

Q: I read from left to right: why aren't inputs on the left and outputs on the right in the formalism?

(Note to question asker, better to use something other than products as product is a structure in some categories)

(Vincent Braunack-Mayer): this is a matter of taste! You can flip the diagrams.

AC: exactly, isomorphic things are the same thing :-)

AC: also note that some people read right-to-left :-)

I guess this is related to the way function composition works. If I want to compose f and g, the output of g has to be the input to f. If I want to write (f o g) then it should look something like
(out--f--in o out--g--in)

Applied Category Theory vs engineering and other fields

Q: Out of curiosity, what are some of the topics niche for mathematics common for engineers?

(Vincent Braunack-Mayer): It gets technical, but things like “profunctors” are quite niche for mathematicians (having taught category theory to math students, these things don’t come up often…)

Q: Maybe I'm getting ahead of myself, but is there a field you feel is a nice complement to Category Theory? Something that neatly describes some situations that Category Theory doesn't apply to?

(Vincent Braunack-Mayer): Category theory is the field of study of (mathematical) structures and how they relate to one another. There is a lot that can be said at the generality of abstract structures (once certain “formal” properties are known) but many “domain-specific” problems (also in maths) require delving into that specific domain. In my experience the answer to this question is quite sensitive to the application

Maybe homotopy type theory? (Nathan McMahon: that can be approached categorically, I think a (1,\infty) category) true but also vice-versa up to a point, no?

VBM: homotopy type theory is related to infinity category theory via a “structure/semantics” correspondence (this is the “higher” or “homotopy” or “infinity” version of the relationship between type theory and category theory)
Something non-algebraic or non-compositional would be a challenge for CT.

Q: Why do some people say that CT can be considered as a theory for the foundation of mathematics? Is this relevant to the course?

(Paul David): CT is generally seen as an umbrella that encompasses almost every area of mathematics. When looking at different domains (algebra, topology, geometry, analysis) we always encounter a type of object (group, manifold, topological space, connected graph) and mappings that preserve that structure (homomorphism, smooth map, continuous map, graph homomorphism, etc).

A category is simply a choice of type of object and type of map. Notice that composition of mappings preserves the structure chosen (a smooth map composed with another smooth map stays smooth). I'm sure this will be answered as the course ensues.

Q. You are right about the "niche" areas of math being attractive to engineers (for me that would be Sheaf Theory). In your experience, which direction do you see engineers going with the "niche" as the starting point? More specialisation to sculpt or architect something from the niche, or to step back and stay at a general level and be content with the "basics"?

AC: more specialization in the direction needed for their field.

Q. How different is category theory from ontology modelling?
Quite!

AC: for the little I know, ontology modeling doesn't map well to categories. Or maybe we will discover that it does if you propose an entry in the encyclopedia.

Good question… Category theory is not concerned with what the objects actually *are* . Everything is encoded in the arrows and how the arrows compose. In category theory you can in essence forget the essence of the objects. :)
In ontology modelling, you are trying to map reality to a formal specification. The essence of an object is essential :)
A domain ontology could describe for example the concepts of mechanics. These mechanical concepts are related somehow and all this is formally specified in an ontology. This is how the world "works" according to mechanics. Or you could describe electrical networks or chemistry.
If you draw an ontology on paper, you draw a diagram and put some words (labels) that have meaning to us. If you keep the drawing the same and change some labels, you might get in trouble… If the ontology of mechanics starts using "spooky action at a distance" and "magic fairy dust" as concepts it makes no sense to us any more. What I mean is, in ontology the words you use matter to us humans even if the drawing itself is a perfectly accurate depiction of what mechanics is.

The thing is though, that there are some formal analogies between engineering ontologies. Masses, friction, pendulum, etc in the mechanics domain are analogous to charges, resistors, RLC, etc in the electrical network domain. That is, you can keep the same diagram and change all the words that describe mechanics to words that describe electrical networks, and it still makes sense. Tada!... a "functor" between mechanics and electrical networks.
https://en.wikipedia.org/wiki/Mechanical%E2%80%93electrical_analogies

Taking the same paradigm one step further, I guess this is what we are doing when we build "mathematical models'' and "theories" in the sciences. We are in essence finding an "analogy" or "functor" from "physical reality" to "mathematical equations''. Eg a "functor" from the physical reality of "electromagnetic fields" to Maxwells equations. And now it turns out that ontological concepts in different domains, that “behave the same way” get mapped to the same mathematical equations, no matter what the original ontological reality was. So what really matters is how the concepts behave and compose, not how they are called.

Maybe the aim of applied category theory is to look at formal analogies between engineering domain ontologies? ...Maybe?… Please comment

AC: Maybe yes if you think of "ontology" philosophically; I was thinking the question was about ontologies in computer science.

About the class experience

Q: How is it planned to create the connections between problems setters and problem solvers?

AC: Two ways to start with:

1) the encyclopedia of engineering categories is a good first step
2) you shall mingle :-) You can put in your Zulip profile if you are a problem solver or a problem haver.

Q: would be great to have a list of participates -- Michael Naderhin

AC: we will make a wiki page where people can list themselves (we didn't ask for permissions to list people). How do people feel about a Linked In group as well?

Response to AC: I would prefer Facebook. Linked In seems more for business and professional. (+1)

Logistics and technical issues

Q: Can the calendar be set up/accessible for people who don’t use Google products? +2
Q - Related: I’m getting the error: “We are sorry, but you do not have access to Google Calendar. Please contact your Organization Administrator for access.” +1

AC: we will see how this can be done.

Q: Can lecture slides be posted before the lecture?

AC: we will post lecture notes before the lectures; in most cases the slides will be "written in" online.

(Not a Q): I think it may be a great idea to have separate sub-groups for undergraduate students, graduate students, postdocs, and people in different fields of industry, which may give people with similar interests an opportunity to connect (+4)

AC: Let us know if you have proposals on how to structure the Zulip streams. Hope there is some self-emergence arising.

Response to AC: I think if you set up separate channels #graduate_student and #postdoc and #engineering, etc. then people can join depending on what area they're in.

Can you do this on Zulip? I'm not familiar with Zulip, but I think it might be possible.

Q: How can we draw those cool diagrams from the slides (I assume is some LaTeX library)?

(Vincent Braunack-Mayer): the tikz package for latex would be my go-to

AC: the very dense ones are created in Tikz by Gioele (ask him to make a public latex package!). The cartoonish ones are done by hand in Keynote.

GZ: Actually, the very dense ones are in Illustrator (gave up on tikz)

Q: Could you please consider using Obsidian (or some other graph note-taking app) as a possible wiki option? (RoamResearch is great too)

Obsidian more individual tool than collective, though. +2
→ If Obsidian is merely a markdown-based note-taking tool, then HackMD does the same thing but allows collaboration.

AC: We will consider it as an option.