Applied Compositional Thinking for Engineers (ACT4E)

Session 10 - Parallelism - Q&A
Questions & Answers

Q: it was a really nice song, where can I download it? https://www.shazam.com/

Q: how does composition work in a monoid? An operation is shown as MxM->M, but then this means we can't compose the operation because first application gave us M whereas we need MxM. As in, given f,g: MxM->M, we can't do f;g because f gave us M whereas g needs MxM

JL: In the definition, M is a *set*, and we define a composition operation (also called "multiplication") which is a *function* * : M x M → M. We think of this as composing *elements* of M, i.e. given elements x, y of, we can multiply them to get x*y, another element of M. Here we are not compositing maps of the form f: M x M --> M. Does that clarify?
OP: I think so, yes, thanks

Q: a follow up question: is a category MxM the same as M? I'm asking because we have a functor F: MxM -> M, so it gives the question similar to previous one.

JL: not sure if this was clear already: in the basic definition of a monoid, we are not yet thinking in terms of categories and functors. M is a set, and multiplication is a function. Subsequently, we start to think of these ideas in terms of category theory…

If M is a category, M x M is not the same as M. Here the "x" symbol is the cartesian product of categories (on the level of objects, the objects of M x M are pairs of objects of M)

In the definition of a monoidal category, the monoidal product is a *functor* M x M → M, in analogy with the definition of monoid above.

Q: The monotonicity requirement for monoidal poset is 'extra' to the general monoid requirements. Can you comment on why this is needed?

JL: the idea is the that in the definition of a “monoidal poset” we don’t just want to put a monoidal structure on the underlying set of the poset, but rather the definition gives a monoidal structure which is *compatible* with the poset structure.

Q: in the example I’m not sure that your monotonicity illustration is correct.? Need probably to add boxes with the product inside...no?

JL: Which slide number / which example?

GO: slide 21 but there is no writing in the slide only “live”, Gioele wrote by hand the illustration of monotonicity wiring diagram . OK I understand thanks!!

JL: Ah, ok, I’m not sure what he wrote :) I guess we’ll see in the recording… we can perhaps go over this later again if there is uncertainty about how it should be.

Perhaps what you are referring to is this rule?
“If x \leq y, and z \leq w, then x * z \lew y * w”

GO: Yes exactly!

JL: Ok, great. We can draw it again in the Q&A session.

GO: Can we have an example of monoidal category “without elements” (as it’s the case for Set)
NM: I think the question is what kinds of categories are there where your objects aren’t sets
YES!

GO: maybe you can consider element of an engineering structure without “natural” elements “inside”...I’m not engineer just I ry to imagine…

JL: Yes, this kind of thing exists. I can think of some math examples without “elements” but no engineering ones off the top of my head right now.

<u>*GO: math example would be ok also!*</u>

JL: Mathy examples that come to mind
- monoidal posets
- rings, viewed as monoidal categories with one object*(NM: I think this would mean that your object is a set and therefore has elements. JL: No, here the object is more like a place-holder; all the “action” happens on the level of morphisms. This is similar to the case where we can think of a monoid M as a category with one object. The object is not the set M, but rather a place-holder object (call it P, say), and we thing of the elements of M as *morphisms* from P to P… so M as a monoid corresponds to Hom(P,P), with the multiplication on M corresponding to the composition operation for morphisms.)

Oh, and in computer science people often work with categories where the objects are “types”. Morphisms then are e.g. processes that have certain types as inputs and other types as outputs.

NM: Resource theories, e.g. canisters of gas at certain pressures and temperatures can be your objects, your morphisms are the physical transformations you can do on these gases or reactions you can produce. (I think these can also be thought of as a **Pos**-enriched category as the existence of a physical transformation from A to B means its feasible to go from A to B)

Remark: Wow! Are we considering braided wiring diagrams? When is this needed in an engineering application?

JL: I would expect most examples to be symmetric monoidal, but think there are examples that are not… e.g. the list monoid is not symmetric. In any case, DP is symmetric monoidal.
KL: And quantum physics
NM: braided categories tend only to show up in quantum physics if you are considering anyons, less esoteric quantum mechanics, e.g. objects as hilbert spaces/qubits/bosons/fermions will end up being symmetric categories. (though topological quantum computing would use braided string diagrams)
Note: you can trace a non-symmetric monoidal category but its a bit more complicated than tracing a symmetric monoidal category.

KL: This got me thinking a bit. Where would you encounter braided diagrams in “real life”? Möbius bands and such came to mind. Spinors... quaternions…Who was it that asked about “geometric algebra”? Maybe relevant with braided stuff when looking at clifford algebras...
https://en.wikipedia.org/wiki/Plate_trick
An engineering application: https://en.wikipedia.org/wiki/Anti-twister_mechanism
Image analysis…https://www.ima.umn.edu/2009-2010/SW10.5-7.09/8058
Robotics...https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwic9JXFwMHuAhXnsosKHSSVB_4QFjACegQIAhAC&url=https%3A%2F%2Fwww2.cose.isu.edu%2F~perealba%2Fdocuments%2FKinSynthesisUofU.pdf&usg=AOvVaw1_STTHCzeJRSWcxPewNp_X
More robotics applications:
https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.138.7333&rep=rep1&type=pdf

JL: The braid group https://en.wikipedia.org/wiki/Braid_group is closely connected to braided monoidal categories… and comes up in knot theory and mathematical physics… but I don’t know this story very well. There are lots of interesting connections between knot theory and physics.

Q: Can we clarify the Pos-enriched category example? What possible operations are available between two parallel morphisms in C? What extra operations are available between two endo-morphisms? Is “and” operator and “tensor” operator the same in this case?

GZ: see recording

GO: Sorry I do not understand what does it mean that Hom(x,y) is in OBJ_bool ? OK Thanks!

JL: in the usual definition of a category, for each pair (x,y) of objects, Hom(x,y) is a *Set.* In the enriched world, e.g. when we enrich using the category Bool, we require that for each pair (x,y) of objects, Hom(x,y) is an object in *Bool.* With this perspective, we
could call “ordinary categories” instead “Set-enriched categories”.