Applied Compositional Thinking for Engineers (ACT4E)

Session 2 - Connection

Questions & Answers

Q. Is there a reason that the category Set has functions as morphisms rather than relations? Why is it that relations are second class citizens when they generalize functions and are perfectly composable?

AC: Well, there is a category of sets and functions, that is called Set.
The category of sets and relations is called Rel.
Not sure either of these are a "first" or "second" class citizen…

DS: The ethos of category theory is that all categories are first-class things. As Andrea says, we can compare them later (e.g. with a functor Set → Rel). Maybe the answer you're looking for is "Yes! Relations are really cool, and they're not second class at all. We love them and want more people to know about them."

GZ: In Session 3, we will see that Set is a *subcategory* of Rel when we introduce the concept.

Q. I find this confusing because my understanding was that categories are fundamentally about functions and that's why de Moor et al talk about an "allegory" instead when they want a category over relations, and added new operators (inclusion, meet and transpose) that aren't required in the standard category.

AC/JL: An allegory is a particular case of 2-category. A 2-category is one where the homsets are categories themselves and there are "2nd order morphisms" between morphisms. We will do as a special case of 2-categories locally posetal categories.

GO: In CT objects have generally no elements… a morphism between two positive integers can be a matrix….

Follow-up: When people name a category after objects (like Set), the morphisms are assumed to be functions (that satisfy some nice comm properties or whatever), no? I guess I'm really thinking about whether functions are "natural" in some way, from a categorical POV.

AC: I don't know if there is a rule. I think it's just a historical accident they called Set the one with sets and functions.

For the record, other fun categories where objects are sets:

- Set (sets, functions)
- Rel (sets, relations)
- Sets with inclusions. There is a morphism f:B->C if B is a subset of C.
- Sets with injective functions (this one has a name I think)

FJR: I suspect it's because category theory emerged from algebraic topology, where they were thinking about functions, so they defaulted to Set having functions as morphisms.

AC: Regarding the question: "are functions "natural" or "special" in Category Theory?" then the answer is probably not at the base level. For example, Rel is nicer than Set. (and we will see that the category of design problems is an even nicer Rel).

However, the concept of *functor* is defined in terms of functions. So you cannot do a lot of category theory if functions don't exist.

How about set-valued functions?

AC: There is certainly a few categories where the morphisms are set-valued functions and the objects are sets. Here's one way to put it:

A morphism f : X -> Y is a function F: X -> PowerSet(Y).
The identity id: X -> X is the map ID: x | ---> {x} (maps x to a singleton)
For composition:

```
   f: X -> Y              g: Y -> Z
-----------------------------------
    f;g: X -> Z
```

the composition can be described as

x -> U_{y \in F(x)} G(y)

Ah, note this is a good example where the morphism f : X -> Y is a function **but** it is not a function from X to Y.

This example can be described more synthetically with monads, which we will mention later in the course.

Q. [DJ] A nitpick about your slide that explains how to define functions in terms of relations. You say for example that the relation must be single-valued for each point in the domain, and then say this corresponds to x=x' => f(x) = f(x'). But this latter formula only makes sense for functions given the assumption that x and x' are mapped to single values. So it's an axiom you can use with functions but it doesn't help define a function, since it could be the case that f(x) is the image of the set {x} under the relation corresponding to the function, in which case the axiom would still hold.

FJR: No, it's fine for relations. It just says that any f(x) has to be the same. Just think of f(x) in relations as a shorthand for "for all f(x) s.t. <x,y> \in R"

[DJ] You mean f(x) is defined to be {y: Y | (x, y) in f}? But then note that the axiom still holds when f is not single valued!

FJR: I was trying to express that you could just not write f(x) and write y instead and introduce appropriate quantifiers. It might be clearer that way.

AC: Yes, in a slide like this:

1. $\forall\, x \in X \quad \exists\, y \in Y \ : \ \langle x, y\rangle \in R_f$

$$\forall\, x \in X\ \exists y \in Y \ : \ y = f(x)$$

"every element of the source X gets mapped by f to some element of the target Y"

2. $\forall\ \langle x_1, y_1\rangle, \langle x_2, y_2\rangle \in R_f$ holds : $\quad x_1 = x_2 \ \Rightarrow\ y_1 = y_2$

$$\forall\ x_1, x_2 \in X \text{ holds : } \quad x_1 = x_2 \ \Rightarrow\ f(x_1) = f(x_2)$$

"f is single-valued"

The black parts are the definition of function.
The blue parts are more like consequences.

DS: I kinda agree with the questioner. The notation f(x) is very "special", i.e. kinda weird from a relational point of view. It bakes in the "single-valuedness".

[DJ] Yes, and note that even in the axiom for totality, the formula

all x | some y | y = f(x)

does not necessarily constrain f to be total, since some treatments of partial functions would allow f(x) to map to an arbitrary value when x is outside the domain of f!

FJR: Maybe dropping the function-like notation is the solution here and writing it infix like x *f* y? I haven't seen the function-like notation for relations for, actually.

DJ: Sure. But if you write it infix, you're back to the relational definition that you already have. Notations for applying relations like functions are common. In Z, for example, r((s)) (should be a special double paren symbol) is the image of the set s under the relation r. In Alloy, this is written s.r (or r[s]), and function application is written x.f (or f[x]), and both are semantically defined as a relational composition (s;r, for example).

NM: Except that the function notation here is a translation so that the reader can see a connection between a subset of relations and functions (and see that the conditions for the relation to be part of this subset corresponds to what you would expect of a function).

FJR: I think I'm actually just wrong. The line on the slide with f(x) is in functions. The line above it is the relation definition. Shouldn't pontificate at 6AM...

Q. (Gregory Paul) It seems that there is a strong relationship between categories and combinatorial objects (graphs, posets, even simplicial/cubical complexes?).

To which extent is it the case, and what kind of techniques/insights from combinatorics are useful in practice in ACT?

Q. In Rel you say that we can compare morphisms, is it something useful in ACT , I never saw that in that sense in "pure" CT?

GZ: It is definitely useful in the category of design problems we will define throughout this course. We will talk about the notion of locally posetal categories, in which you can order morphisms in posets.

NM: I think this example might be related to higher categories in pure CT (2-categories or bi-categories)

Q: Is it possible that the morphisms of a category are not even relations?

AFAIR morphisms can be categories themselves. (Correct me if I'm wrong please.) But that's really advanced stuff.
NM: That is a 2-category, related to the idea of enriched categories which answers this question, where each Hom-set (morphisms between a pair of objects) is itself an object from a fixed monoidal category (for 2-categories the monoidal category is the category of all small categories).

NM: one example is that the morphisms can be functors.
Yeah, but that's not that much different from functions, which are special cases of relations.
Another example would be numbers, you could choose to enrich your category by the integers and then composition is addition. (I can't say when you would use that exactly though)

Q. Do you see or have experienced fruitful interactions of Mereology with ACT, for engineering?

In [Seven Sketches in Compositionality](), Chapter 7.3. Sheaves:
But if we think of R_2 as a site where things happen, then we might think of things like weather systems throughout the plane, or sand dunes, or trajectories and flows of material.

Q. Can you give 2 or 3 examples of categorical constructions like product/coproduct/SLice category/etc… that are relevant in ACT/Engineering ?

NM: symmetric monoidal categories are relevant to resource theories (such as energy transformations).

GZ: Traced-monoidal categories are relevant to the category of design problems, which we will discuss in this course

Q. Can we have a "cheat sheet" for all the mathematical notations used in this course?

GZ: yes, we will add something in that spirit in the book

Q. Would it be useful to have a stream on Zulip where we can express our interest in defining a particular category? Maybe we can find teams to work on a category.

GZ: Sure, that's a good way to meet people with interests aligned to yours. See AC's zulip's post for more information.

Maybe stream "Categories Encyclopedia"

Q. What can we do with relations that we can't do with functions?

AC: for example, "going backwards".

Q. As an electrical power engineer, I am a bit confused by the first example. Why the causality? How can you determine it? The conception of an electrical network is a set of generators that inject current into a network from which customers extract it - similar to a hydraulic network. There's no possibility of tracing a specific customer to a specific generator. What type of insight would this abstraction give me?

NM: I personally don't have a good answer for this (not one of the instructors), but perhaps having this kind of construction/definition means you can use the general design problem solutions essentially "out of the box"