Categories as I came to know them

James B. Wilson

2024-01-25

These days when I think of a category I see a monoid with holes and I think for some of you it might help to know this perspective as well. So what do I mean by a monoid?

A *monoid* is a multiplication $g \cdot h$ and an identity 1 such that

$$g \cdot (h \cdot k) = (g \cdot h) \cdot k \qquad\qquad 1 \cdot g = g = g \cdot 1.$$

These are signatures of a monoid not the required notation. You may think of the natural numbers

$$n : \mathbb{N} = 0 : \mathbb{N} \text{ or a successor } S(k) : \mathbb{N} \text{ of an existing } k : \mathbb{N}.$$

Successors are for counting—a tally, but I often count in groups and add the results so this gives over to the common notion of addition:

$$m + n = \begin{cases} m & n = 0 \\ S(m+k) & n = S(k) \end{cases}$$

In this monoid the role of $\cdot$ is played by $+$ and 1 by 0. Meanwhile if I switch to using multiplication of natural numbers

$$m \cdot n = \begin{cases} 0 & n = 0 \\ m \cdot k + m & n = S(k) \end{cases}$$

I still get a monoid only now $\cdot$ is the product and 1 is $1 : \mathbb{N}$.

Now an *category* is the same idea as a monoid only we accept that sometimes products we are not everywhere defined.

Why would a product not be everywhere defined?

Some products are too expensive to define everywhere. Say what you will about the existence of every natural number, but there is no point in multiplying two random 1 million digit integers. Even if it is possible it is an unforgivable waste of energy, time, and money. That limit tends to break many would-be monoids. As a pint-size example try the sum and product of natural numbers $0,\ldots,15$—a traditional engineering limit imposed on some of the first computers which got the name *nibble* in the 1950's.Those numbers got to be written in Hexidecimals

$$H = \{0,\ldots,9,A,B,C,D,E,F\}.$$

I.e. $C = 12$. The arithmetic operations the addition and multiplication tables would be these.

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
2	3	4	5	6	7	8	9	A	B	C	D	E	F		
3	4	5	6	7	8	9	A	B	C	D	E	F			
4	5	6	7	8	9	A	B	C	D	E	F				
5	6	7	8	9	A	B	C	D	E	F					
6	7	8	9	A	B	C	D	E	F						
7	8	9	A	B	C	D	E	F							
8	9	A	B	C	D	E	F								
9	A	B	C	D	E	F									
A	B	C	D	E	F										
B	C	D	E	F											
C	D	E	F												
D	E	F													
E	F														
F															

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	2	4	6	8	A	C	E								
0	3	6	9	C	F										
0	4	8	C												
0	5	A	F												
0	6	C													
0	7	E													
0	8														
0	9														
0	A														
0	B														
0	C														
0	D														
0	E														
0	F														

There are gaping holes in these two operations so $+$ and $\cdot$ are not even functions $A \times A \to A$ so they cannot be operations in that technical sense.

Now for some reality. Computers actually do something when asked to add or multiply beyond the bounds. Most output some sort of error code, something like OVERFLOW. I will use $\perp$. So in that model computers and I are extending H to include the error outcome like this

$$H^? = H \sqcup \{\perp\} = \{0,\ldots,9,A,\ldots,F,\perp\}.$$

So our addition and multiplication tables are now functions $+,\cdot : H \times H \to H^?$ or what are often called *partial* operations because of the possible error state. This at least makes it clear that we have functions back. What about our monoids?

This sort of failure does break the monoid conditions on addition and multiplication but only a mostly uninteresting way. For example we could extend our sum and product by the *absorption rule*

$$n + \perp = \perp = \perp + n \qquad\qquad n \cdot \perp = \perp = \perp \cdot n.$$

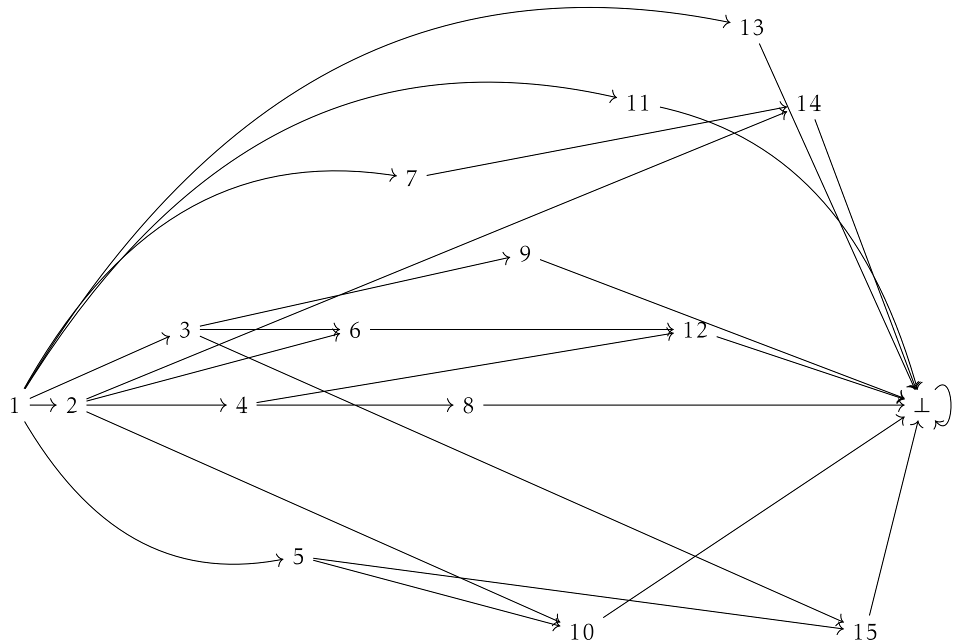
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	⊥
1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	⊥	⊥
2	3	4	5	6	7	8	9	A	B	C	D	E	F	⊥	⊥	⊥
3	4	5	6	7	8	9	A	B	C	D	E	F	⊥	⊥	⊥	⊥
4	5	6	7	8	9	A	B	C	D	E	F	⊥	⊥	⊥	⊥	⊥
5	6	7	8	9	A	B	C	D	E	F	⊥	⊥	⊥	⊥	⊥	⊥
6	7	8	9	A	B	C	D	E	F	⊥	⊥	⊥	⊥	⊥	⊥	⊥
7	8	9	A	B	C	D	E	F	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
8	9	A	B	C	D	E	F	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
9	A	B	C	D	E	F	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
A	B	C	D	E	F	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
B	C	D	E	F	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
C	D	E	F	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
D	E	F	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
E	F	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
F	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	⊥
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	⊥
0	2	4	6	8	A	C	E	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
0	3	6	9	C	F	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
0	4	8	C	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
0	5	A	F	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
0	6	C	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
0	7	E	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
0	8	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
0	9	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
0	A	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
0	B	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
0	C	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
0	D	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
0	E	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
0	F	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥

This adaptation restores the properties of a monoid because the addition and the multiplication are now associative.

In general $\mathbb{N}/_{k=k+1}$ allows for a similar approach for any k. It is common to use diagrams that depict how these data could be generated.

$$\begin{tikzcd} 0 \arrow[r] & 1 \arrow[r] & 2 \arrow[r] & 3 \arrow[r] & 4 \arrow[r] & 5 \arrow[r] & 6 \arrow[r] & 7 \arrow[r] & 8 \arrow[r] & 9 \arrow[r] & A \arrow[r] & B \arrow[r] & C \arrow[r] & D \arrow[r] & E \arrow[r] & F \arrow[r] & \perp \arrow[loop right] \end{tikzcd}$$



The second way that monoid-type products could fail to be defined is when there was no intentional monoid behind it in the first place. Suppose I give you instructions to walk. Walk 2 block north, then 3 blocks west, then wait there. That is a partial addition. I am allow you to follow north walking by west walking. There is no reason to believe from those instructions that you could combine the north with itself.

$*$	e	h	f
e	e	$\perp$	$\perp$
h	h	$\perp$	$\perp$
f	$\perp$	h	f

(0.1)

However if we do not include that case then

So the addition of In this case the number $1 := S(0)$ is not the identity but rather it generates every term $n : \mathbb{N}$ as a monoid. "Generates as a monoid" is jargon to say that $n = f(1)$ for some $\{\cdot, 1\}$-formula $f(x)$. So we say that $\langle \mathbb{N}, +, 0\rangle$ is a *cyclic monoid*. Another monoid on $\mathbb{N}$ uses multiplication $m \cdot n = 0$ if $n = 0$, or $m \cdot k + m$ if $n = S(k)$ and here 1 is the identity. In that monoid every n is the evaluation of a $\{\cdot, 1\}$-formula $f(x_2, x_3, x_5, x_7, \ldots)$ where the indicies run over the primes and $n = f(2, 3, 5, 7, \ldots)$ is to write n as a product of its prime factors. So it is the operation more so than the type of data the dictates how complicated the algebra really is.

As a consequence our identity 1 may split into a collection $\mathbb{1}$ of many identities. Since our products might not be defined we replace equals (=) with "equal when both sides defined" ($\asymp$) and thus to write down a category will mean to describe a product with the following two rules.

$$g \cdot (h \cdot k) \asymp (g \cdot h) \cdot k \qquad\qquad \mathbb{1} \cdot g = \{g\} = g \cdot \mathbb{1}.$$

We can be more precise about this because in a category we will insist that we are warned when a product will not exist so that we may plan around errors instead

The second way that natural numbers from a monoid is to use $e = 1$ and the following common multiplication of natural numbers.

$$m \cdot n = \begin{cases} 0 & n = 0 \\ m \cdot k + m & n = S(k). \end{cases}$$

Here the numbers $n : \mathbb{N}$ are a product of primes so each n is equal to $f(P)$ where $f(X)$ is a $\{\cdot, e\}$-formula over a countably infinite set X of variables. If this monoid were finitely generated then there would finitely many primes which is false. So the negative is true $\langle \mathbb{N}, \cdot, 1 \rangle$ is not finitely generated.

Notice in particular that the condition of finite and infinite generation is not a quality of cardinality of the sets here are the same.

and find these qualities. The monoid of natural numbers under addition uses 0 as e and $+$ as $\cdot$ and is generated by $1 = S(0)$ An *abstract category* is a the same only the multiplication may not be partial and likewise the identity may split into may parts.

partial product that is associative and has an identity.

In the beginning there were functions and functions had composition which was associative. Then some thought that there should be types and functions were divided into types $f : X \to Y$

and composition became untenable with types. So composition was declared restricted. No longer could all functions compose but only those $f : X \to Y$ and $g : A \to B$ where $A = Y$. What would equal types mean? No one was certain but those who insisted on this rule insisted they could continue even if that was not clear. And so it became the rule across the land that composition of typed functions would obey typed composition.

Given two types X and Y and an untyped function f, we if for every $x : X$, $f(x) : Y$ then we say that we may introduce a typed function also denoted by f of type $X \to Y$. Define

$$Y^? = Y \sqcup \{\perp\}$$

where we can think of $\perp$ as a symbol for an error. Given an untyped function f where $x : X$ yields $f(x) : Y \sqcup \{\perp\}$, then we say f introduces a *partial typed function* and denoted $f : X \to Y^?$.

An *abstract category* is a type A equipped with a binary partial operation

Guarded operation

How do you invert an integer n? You write $1/n$. That is of course unless $n = 0$, at which point we do not bother to define an inverse. So there is a guard $\lhd n = n$ as a function $\lhd : \mathbb{N} \to \mathbb{N}$ such that

$$n^{-1} = \text{ if } \lhd n = 0 \text{ then } \perp \text{ else } 1/n.$$

We could extend this guard to all rational numbers

$$\lhd \frac{m}{n} = m$$
$$\left(\frac{m}{n}\right)^{-1} = \begin{cases} \perp & \lhd \frac{m}{n} = 0 \\ \frac{n}{m} \in \mathbb{Q} & \text{else.} \end{cases}$$

If we want we could think of a function $\square^{-1} : \mathbb{Q} \to \mathbb{Q}^?$. This is place where filling the hole with anything that behaves as a number would be to disrupt the intended algebra. So it is a hole we wish to maintain.

Now to amend my earlier claim, a category is a monoid with **guarded** holes. The operation that can have holes is the binary one and so the guard now needs to be on the left and the right. Now the partial axiom has chirality based on whether we think of $\cdot$ as to be read left-to-right ($\triangleright$) or right-to-left ($\triangleleft$). In order to abbreviate a digression into the topic of grammars and signatures, I will write $\square$ for the position of an input to function.

$$\square \cdot \square : C \times C \to C^{?} \tag{Product}$$

$$\triangleleft\square : C \to C \qquad \square\triangleright : C \to C \tag{Source}$$

$$\square\triangleleft : C \to C \qquad \triangleright\square : C \to C \tag{Target}$$

Guards are always total functions.

$$x\triangleleft = \triangleleft y \Rightarrow x \cdot y \in C \tag{Left2Right}$$

$$x\triangleright = \triangleright y \Rightarrow x \cdot y \in C \tag{Right2Left}$$

Now the axioms. In order to state the associative law we need to consider that $x, y, z \in C$ where each of $x \cdot y$, $y \cdot z$, $x \cdot (y \cdot z)$ and $(x \cdot y) \cdot z$ are defined. We should like this occur whenever only the first two are defined and so we need some assumptions that could lead there. One such choice made perhaps by Freyd is the following.

$$\triangleleft(x \cdot y) = \triangleleft(x \cdot \triangleleft y)$$