Categories as I came to know them

James B. Wilson

2024-01-18

These days when I think of a category I see a monoid with holes. What do I mean? 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 role of $\cdot$ is played by $+$ and 1 by 0. Meanwhile if I switch to using multiplication of natural numbers

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

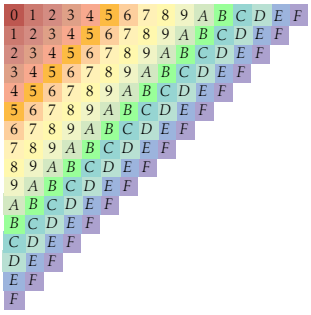
I still get a monoid only now $\cdot$ is the product and $1 : \mathbb{N}$ is in fact the 1. Before continuing on to categories it is worth pausing to notice

how different these two algebraic structures really are. In $\langle \mathbb{N}, +, 0 \rangle$ every number $n : \mathbb{N}$ is $f(1)$ for a $\{\cdot, 1\}$-formula $f(x)$ in one variables, e.g. $3 = (1 + 1) + 1 = ((x \cdot x) \cdot x)|_{x:=1}$ when we replace products by $+$. Meanwhile to write that same n as a true product of natural numbers would need access to all the primes, i.e. $6 = 2 \cdot 3$ and $35 = 5 \cdot 7$.

Now an *(abstract) category* is the same idea as a monoid only we accept that sometimes products we are not everywhere defined. There are two ways this could happen. One is that our imagination gets ahead of Engineering. The obvious examples are from computation where ultimately the physical bounds limit what we may compute. Take for instance, the sum of natural numbers $0, \ldots, 15$—a traditional Engineering limite known as a *nibble*, 4 bits, and often written with Hexidecimals

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

The arithmetic operations the addition and multiplication tables would be these.



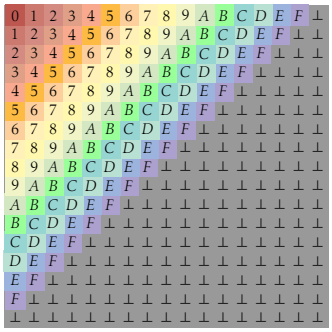
There are gaping holes in these two operations. Most computers fill these with an error signal something like OVERFLOW. I will use $\perp$. So in that model I could extends H to include the error outcome like this

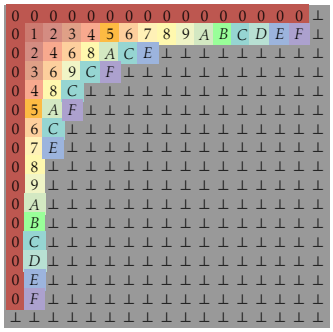
$$H^? = H \sqcup \{\perp\} = \{0, \ldots, 9, A, \ldots, F, \perp\}.$$

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 sort of failure does break the monoid

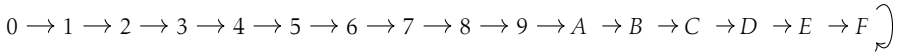
conditions on addition and multiplication but only a mostly uninteresting way. For example we could extend our sum and product by the *absorption rule*

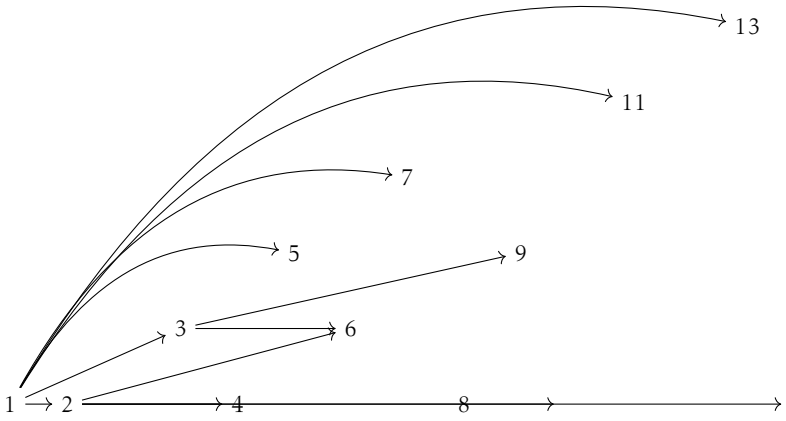
$$n + \bot = \bot = \bot + n \qquad n \cdot \bot = \bot = \bot \cdot n.$$





This adaptation restores the properties of a monoid, the addition and the multiplication are now associative. The perhaps disputable position is if $0 \cdot \bot$ should be $\bot$ or 0 but since we think of $\bot$ originally as an error condition we think it best to continue reporting the error. After all $0 \cdot n = 0$ is not an axiom of monoids so nothing is violated here. These monoids are in fact well-known an extend beyond Hexidecimals. 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.





In this way computers turn their nonassociative operations back into associative ones without requiring deep conceptual leaps.

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.

$*$	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

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

where we can think of $\bot$ as a symbol for an error. Given an untyped function f where $x : X$ yields $f(x) : Y \sqcup \{\bot\}$, 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