UNITS, GENERATORS, AUTOMORPHISMS

Justin T. Chun, December 2021

This protoknole assumes the content of "Name and Number."

How the integers divide

Before anything, there is nothing.

We denote by 0 the **origin**.[1] The origin gives us the unapologetically inert **identity**, denoted by $x \mapsto x$. Its sole purpose is to absolutely nothing.

After nothing, there is something.

We denote by 1 the **unit**. The unit also gives us the **increment**, the first iterative function, denoted by $x \mapsto x^{++}$. We define $0^{++} = 1$, and induct to get the rest:

$$1^{++}, (1^{++})^{++}, ((1^{++})^{++})^{++}, (((1^{++})^{++})^{++})^{++}$$

We'll give those things names in a little bit. Right now, the point is that you can increment one as many times as you want. The image of the increment function is called the **successor**. For instance, 1 is the successor of 0.

When something truly is, something else truly isn't, yet neither are truly nothing.

We denote by -1 the **antipode**. The antipode gives us **negation**, the first involution, defined by $x \mapsto -x$. The initial values are $-0 = 0$ and $-1 = 1$. We can also negate the increment, to get the **decrement** function $x \mapsto x^{--}$, whose image is called the **predecessor**. Involutions are functions that do nothing after two applications. Thus, $-(-x) = x$. We also have to define $(-1)^{++} = 0$, so that

$$((-1)^{++})^{++} = 0^{++} = 1 = (1^{++})^{--} = (((1^{++})^{++})^{--})^{--}.$$

Great, now the notation gets messy twice as quickly. Let's fix this. The integers in bold are the ones defined so far:

$$\ldots, -9, -8, -7, -6, -5, -4, -3, -2, \mathbf{-1}, \mathbf{0}, \mathbf{1}, 2, 3, 4, 5, 6, 7, 8, 9, \ldots$$

We'll call the sequence

$$3,\ 5,\ 7,\ 11,\ 13,\ 17,\ 19,\ 23,\ 29,\ 31, \ldots$$

the **primes** and the sequence

$$2,\ 4,\ 8,\ 16,\ 32,\ 64,\ 128,\ 256,\ 512,\ 1024, \ldots$$

the **seconds**.

[1] Note for nerds: the von Neumann construction of the integers defines $0 = \varnothing$. We don't need this, though it is quite satisfying, since you can talk about every number as a set. For example, x^{++} can be defined as $x \cup \{x\}$.

The primes are very difficult to predict, while the seconds almost have a rhythm to them. The number 2 gives us the **doubling** function $n \mapsto 2n$. We define $2(0) = 0$ and $2(1) = 2$. Note that doubling 1 gets us the image of 2 under the identity map, and that

$$2(2(1)) = 2(2) = 4, \quad 2(2(2(1))) = 2(4) = 8, \quad 2(2(2(2(1)))) = 2(8) = 16, \quad \dots$$

In fact, each second gives as another function. For example, 4 gives us the **quadrupling** function; the initial values are $4(0) = 0$ and $4(1) = 4$, and the rest goes something like

$$4(4(1)) = 16, \quad 4(4(4(1))) = 4(16) = 64, \quad 4(4(4(4(1)))) = 4(64) = 256, \quad \dots$$

Taking it a step further, 8 gives us the **octupling** function, where the initial values are $8(0) = 0$ and $8(1) = 8$ and it continues on in this fashion:

$$8(8(1)) = 64, \quad 8(8(8(1))) = 8(64) = 512, \quad 8(8(8(8(1)))) = 8(512) = 4096, \quad \dots$$

Something really nice is happening here: we can write the seconds in terms of each other in a really concise way by just composing functions such as double and quadruple. To prove my point, you can check that

$$35184372088832 = 512(256(128(64(32(16(8(4(2(1)))))))))).$$

Let's go back to the primes. We can play a similar game here.

The number 3 gives us the **tripling** function $n \mapsto 3n$. The initial values are $3(0) = 0$ and $3(1) = 3$, and it continues

$$3(3(1)) = 9, \quad 3(3(3((1)))) = 27, \quad 3(3(3(3(1)))) = 81, \quad \dots$$

The number 5 gives us the **quintupling** function $n \mapsto 5n$. The initial values are $5(0) = 0$ and $5(1) = 5$, and it continues

$$5(5(1)) = 25, \quad 5(5(5(1))) = 125, \quad 5(5(5(5(1)))) = 625, \quad \dots$$

We can also adapt this algorithm to accomodate involutions such as negation. For example:

$$-1(-1(1)) = 1, \quad -1(-1(-1(1))) = -1, \quad -1(-1(-1(-1(1)))) = 1, \quad \dots$$

Also, with initial values $1(0) = 0$ and $1(1) = 1$ and $0(0) = 0$ and $0(1) = 0$, we have

$$1(1(1)) = 1, \quad 1(1(1(1))) = 1, \quad \dots \quad \text{and} \quad 0(0(1)) = 0, \quad 0(0(0(1))) = 0, \quad \dots$$

Time to break the ice: this is **multiplication**.[2] Oh, and **addition** is how we keep track of iterated increments and decrements:

$$1 + 1 = 1^{++} = 2, \quad 1 + 2 = 1 + 1^{++} = 1 + 1 + 1 = 1^{++} + 1 = 2 + 1, \quad -1 + 1 = 0, \quad \dots$$

Now that we have a few functions that go in different directions, it may be worth asking which ones are reversible. We call such a reversal an **inverse function**. For example, it should be clear that the inverse function of the increment is the decrement.

[2] It works with negative integers, too. I just don't feel like writing it out.

The inverse function of addition always exists (because you can always reverse a series of increments using a series of decrements) and is called **subtraction**. Also note that negation is its own inverse (indeed, any involution has this property; this is why the fancy name "involution" is warranted).

The inverse of multiplication gives rise to an entire branch of mathematics called number theory and is called **division**. Over the integers, most numbers aren't divisible by most other numbers. For example, the only numbers with multiplicative inverses are 1 and -1. Multiplicatively inverting 0 is usually forbidden for reasons that I don't feel like getting into (basically, the space collapses in on itself).

When division is valid, we say that the number being divided is called the **dividend** (or numerator) and the number doing the dividing is called the **divisor** (or denominator). We call a divisor of a number **proper divisor** if that divisor is both not equal to that number and not 1. The primes have zero proper divisors, whereas the seconds have as many proper divisors as can be had.

Multiplication tables

So, 1 and -1 are special among the integers. As such, they are considered **multiplicative units**.[3] Since anything times 0 is 0, we call 0 a **multiplicative annihilator**.[4] Let's organize the data in a neat little table:

$\cdot$	-1	0	1
-1	-1	0	1
0	0	0	0
1	1	0	1

If you're still reading, you're probably familiar with the Gaussian integers $\mathbf{Z}[i]$. Here's what the multiplication table looks like there (0 is omitted since it's so predictable):

$\cdot$	$-i$	-1	1	i
$-i$	-1	i	$-i$	1
-1	i	1	-1	$-i$
1	$-i$	-1	1	i
i	1	$-i$	i	-1

Note that the symmetry along the NW-SE diagonal tells us that though we relinquish total order over the analytic numbers, multiplication over $\mathbf{Z}[i]$ is still commutative. This is because we're only really introducing one new arithmetic rule when we work over $\mathbf{Z}[i]$, namely, that $i^2 = -1$.

Also note that the units group here fits neatly into quadrants of the multiplication table, i.e. we could've written out half of the table and the no real information would be lost (though some would argue that it would be a tad harder to read).

[3] If you know some algebra, this is the simplest **units group**, which makes sense since $\mathbf{Z}$ is the simplest ring.
[4] 0 is also an **additive unit**, but since addition is so nice, no one bothers saying this.

The quaternions are the first number system we encounter where inspecting the unit group provides actual insight into what would otherwise be an opaque realm. We have the rules

$$ij = k, \quad jk = i, \quad ki = j \quad \text{or, alternatively,} \quad i^2 = j^2 = k^2 = ijk = -1.$$

The i here behaves exactly how it does in $\mathbf{C}$. Here's half of the multiplication table, with zero omitted again:

1	$-i$	j	$-k$	k	k	$-j$	i	-1
i	1	$-k$	$-j$	j	j	k	-1	$-i$
$-j$	k	1	$-i$	i	i	-1	$-k$	j
$-k$	$-j$	$-i$	-1	1	1	i	j	k
$-k$	$-j$	$-i$	-1	$\cdot$	1	i	j	k

Note that multiplication over the quaternions is not commutative. This means the order with which one reads the above table matters.[5]

Permuting the axes

Now, check out how two quaternions multiply:

$$\begin{aligned}
&(A_0 + R_0 i + G_0 j + B_0 k)(A_1 + R_1 i + G_1 j + B_1 k) \\
&\quad = A_0(A_1 + R_1 i + G_1 j + B_1 k) + R_0 i(A_1 + R_1 i + G_1 j + B_1 k) \\
&\quad + G_0 j(A_1 + R_1 i + G_1 j + B_1 k) + B_0 k(A_1 + R_1 i + G_1 j + B_1 k) \\
&\quad = A_0 A_1 + A_0 R_1 i + A_0 G_1 j + A_0 B_1 k \\
&\quad + R_0 A_1 i - R_0 R_1 + R_0 G_1 k - R_0 B_1 j \\
&\quad + G_0 A_1 j - G_0 R_1 k - G_0 G_1 + G_0 B_1 i \\
&\quad + B_0 A_1 k + B_0 R_1 j - B_0 G_1 i - B_0 B_1 \\
&\quad = (A_0 A_1 - R_0 R_1 - G_0 G_1 - B_0 B_1) \\
&\quad + (A_0 R_1 + R_0 A_1 + G_0 B_1 - B_0 G_1) i \\
&\quad + (A_0 G_1 + G_0 A_1 + R_0 B_1 - B_0 R_1) j \\
&\quad + (A_0 B_1 + B_0 A_1 + R_0 G_1 - G_0 R_1) k
\end{aligned}$$

Setting $A_0 = A_1 = 0$ recovers the inner product of 3-vectors in the A component and the cross product of 3-vectors in the R, G, and B components.

[5] I'm reading it file then rank, like in standard tournament chess notation. A fun exercise to see how well you track the sign changes is to modify this table so that it reads rank then file. More generally, it is good practice to permute through any "canonical" configuration; it aids flexibility when reading historical texts where the practice might not align entirely with current idioms.

I like to think of A as a value sink and R, G, and B as axes in a color space. In particular, singling in on a certain color (like applying a filter in a photo app) is tantamount to choosing a tangent direction for an intrinsic frame. Say we choose R as our tangent axis. If our axes are permuted in a cyclic manner, G would be the normal axis and B would be the binormal axis.[6]

Generators and free groups

Let's collect all those details about unit groups and rings. There is an invertible functor from the category of groups to the category of rings. From left to right, we can throw the unit group into a **free group** generator and get a ring.[7] From right to left, we can take that same ring and perform a distillation, yielding a unit group. For bookkeeping reasons, we'll allow the antipode to move freely throughout this configuration. Thus:

- $\{1\}$ is enough to generate all of $\mathbf{Z}$.
- $\{1, i\}$ with the restriction $i^2 = 1$ is enough to generate all of $\mathbf{Z}[i]$. We need a little more work to get from $\mathbf{Z}[i]$ to $\mathbf{C}$; however, it really isn't the focus of this protoknole.[8]
- In a similar manner, you can generate $\mathbf{H}$ up to some small stuff using $\{1, i, j, k\}$ and the restriction $i^2 = j^2 = k^2 = ijk = -1$.

The point is that you don't need to say much in order to describe the algebra of well-known rings. In this sense, groups are the unit brokers of the ring economy, generating revenue via (usually orthogonal) spanning bets.

Groups and automorphisms

So, is every group just a group of units waiting to be turned into a ring? Well, kinda, but this isn't a very productive way to think about them. A healthier perspective is to view groups as a way of cataloguing **automorphisms** of a geometric object. So, for example, -1 is always part of the conversation with unit groups because you can always flip an axis and end up with a space that is identical algebraically (though usually different analytically). All that talk about i, j, and k permuting the axes over $\mathbf{H}$ is good practice for when your space has 35184372088832 dimensions instead of just four.

[6] Question: what would change if we managed things in an anticyclic fashion? What would we have to adjust in order to choose A as a tangent direction?

[7] Look up what a free group is if you don't already know.

[8] In short, figure out how fractions ought to work and then complete the Cauchy sequences. It's whatever.