INTRODUCTION TO COMPUTER SCIENCE · WEEK 7 · STATION III

Uncountability

Vast is still finite. Now we count things that aren't — and discover that some infinities are bigger than others.

MEANING

What we want to talk about is not countable.

f1	1	1	0	1	1	0	0	0
f2	1	1	1	1	0	1	1	1
f3	0	1	1	1	0	1	1	1
f4	1	1	0	1	1	0	0	1
f5	0	1	1	0	0	0	0	1
f6	1	1	1	0	1	1	0	0
f7	0	1	1	0	1	1	1	0
f8	0	1	1	1	0	1	1	1
flip	0	0	0	0	1	0	0	0

differs from every row — so it is on no row

the diagonal

Consider all infinite sequences of bits: 0110100011… forever. Each one is a complete answer to an endless list of yes/no questions — a behaviour, a function, a real number, a fate.

Claim: no list can contain them all. Not a long list. Not an infinite list. No list.

Cantor's proof is three lines and it is the most consequential argument of the 20th century. We run it twice, slowly.

THE OBJECTS

A subset of the numbers is one yes-or-no answer per number.

	1	2	3	4	5	6	7	8	…
{ }	0	0	0	0	0	0	0	0	…
all	1	1	1	1	1	1	1	1	…
evens	0	1	0	1	0	1	0	1	…
primes	0	1	1	0	1	0	1	0	…
{3}	0	0	1	0	0	0	0	0	…
no name	1	1	0	1	0	1	1	0	…

and almost every subset is this one

subsets as answer sheets · 1 = in

Take any collection of counting numbers — the evens, the primes, just {3}, none of them, all of them. Each one is a subset. To pin one down, walk 1, 2, 3, … and answer in or out, forever.

That answer sheet is an infinite bit string, and every infinite bit string is one. All of them together make the power set, written $2^{\mathbb{N}}$ — two choices, made $\mathbb{N}$ times.

A handful have names. Almost all are an endless coin-flip, with nothing shorter to say about them than the flips.

THE PROOF · 1 OF 2 · SUBSETS

Hand me a list of every subset. I will build one that is not on it.

	1	2	3	4	5	6
S1	○	●	●	●	○	●
S2	●	●	○	○	○	○
S3	○	○	●	●	○	○
S4	○	○	○	○	●	○
S5	●	●	●	○	○	●
S6	○	○	●	●	●	○
D	●	○	○	●	●	●

D disagrees with row n about n

● in · ○ out

Suppose the subsets could be listed: `S1, S2, S3,` …, every single one of them somewhere on the list.

Go down the diagonal and ask each row about *its own number*. Is 1 in S1? Is 2 in S2? Is n in Sn?

Now build `D` by answering the opposite every time: `D = { n : n is not in Sn }`. Nothing exotic — a rule anyone can apply, one number at a time.

`D` is not S1: they disagree about 1. Not S2: they disagree about 2. Not Sn for any n whatsoever. So the list left something out — and it was any list at all.

The same move, on the numbers between 0 and 1.

r1 = 0.	1	4	1	5	9	2	…
r2 = 0.	7	1	8	2	8	1	…
r3 = 0.	2	5	5	5	5	5	…
r4 = 0.	3	3	3	3	3	3	…
r5 = 0.	0	0	0	0	5	0	…
r6 = 0.	4	1	4	2	1	3	…
x = 0.	5	5	4	5	4	5	…

x differs from row n in place n

any list · and the number it missed

Suppose you could list them: `r1, r2, r3, …`, each written out as an endless decimal. Take the first digit of the first, the second of the second, the n-th of the n-th. The diagonal again.

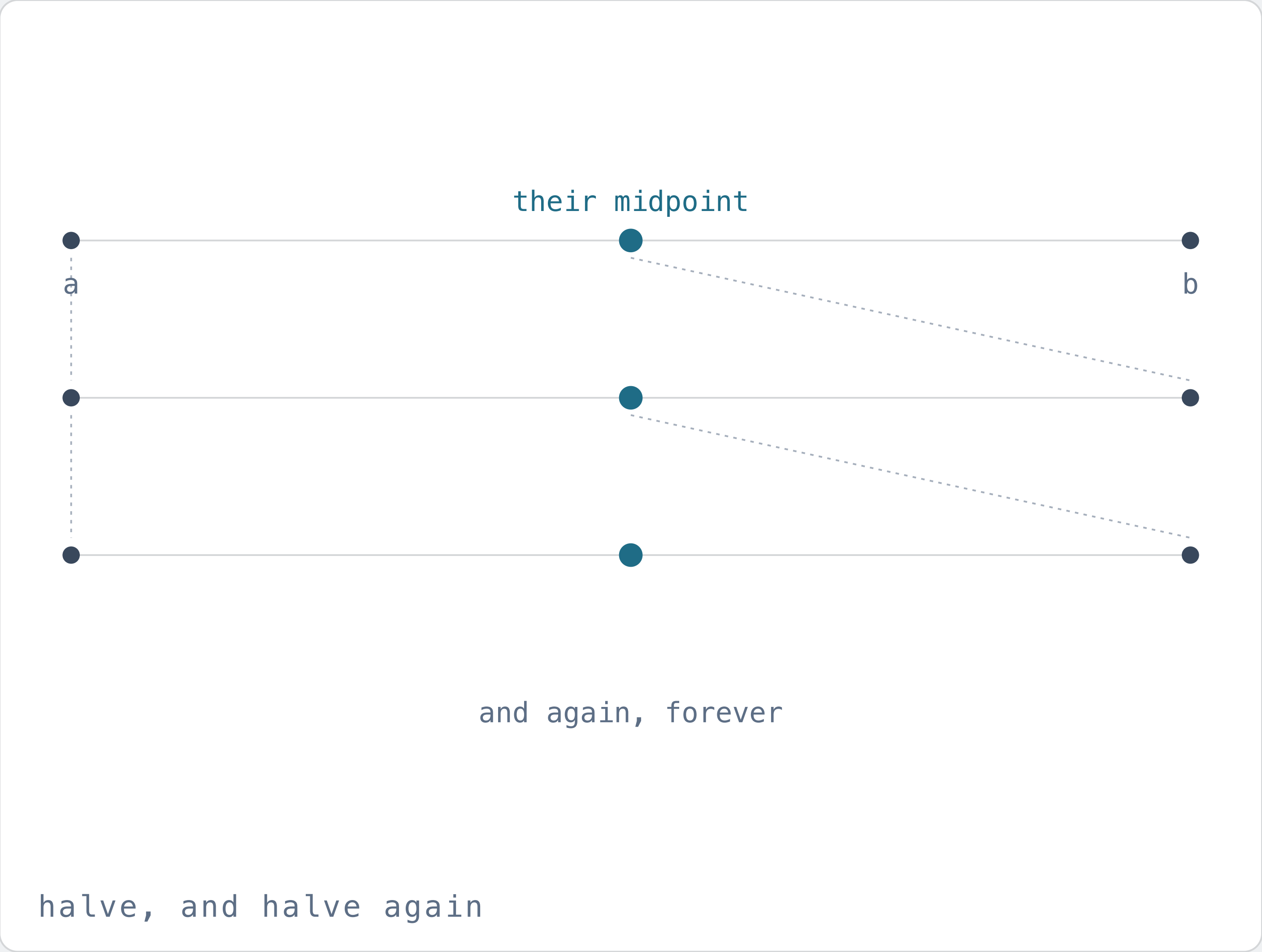
Build a new number x by changing every one of those digits. Then x differs from r1 in the first place, from rn in the n-th — so x is on no row.

THE ONE PLACE IT NEEDS CARE

Change each digit to 5, or to 4 if it was already 5. That keeps you clear of the trailing nines: 0.4999… and 0.5000… are the same number written twice, and a proof that only landed on that would have proved nothing.

INTUITION

Between any two numbers, however close, there is another. But do not let it do the work.



Pick two reals as close together as you like. Their midpoint lies strictly between them. Do it again, forever. There is no *next* real number, and a list is nothing but firsts and nexts.

Density is the feeling, not the reason. The fractions are dense in the same way — and they can still be listed, by walking a grid of numerators and denominators corner by corner. Last week's exercise.

Only the diagonal tells the two cases apart. When an intuition and a proof agree, check which of them you are actually relying on.

THEOREM

The diagonal, once and for all.

CANTOR · 1891

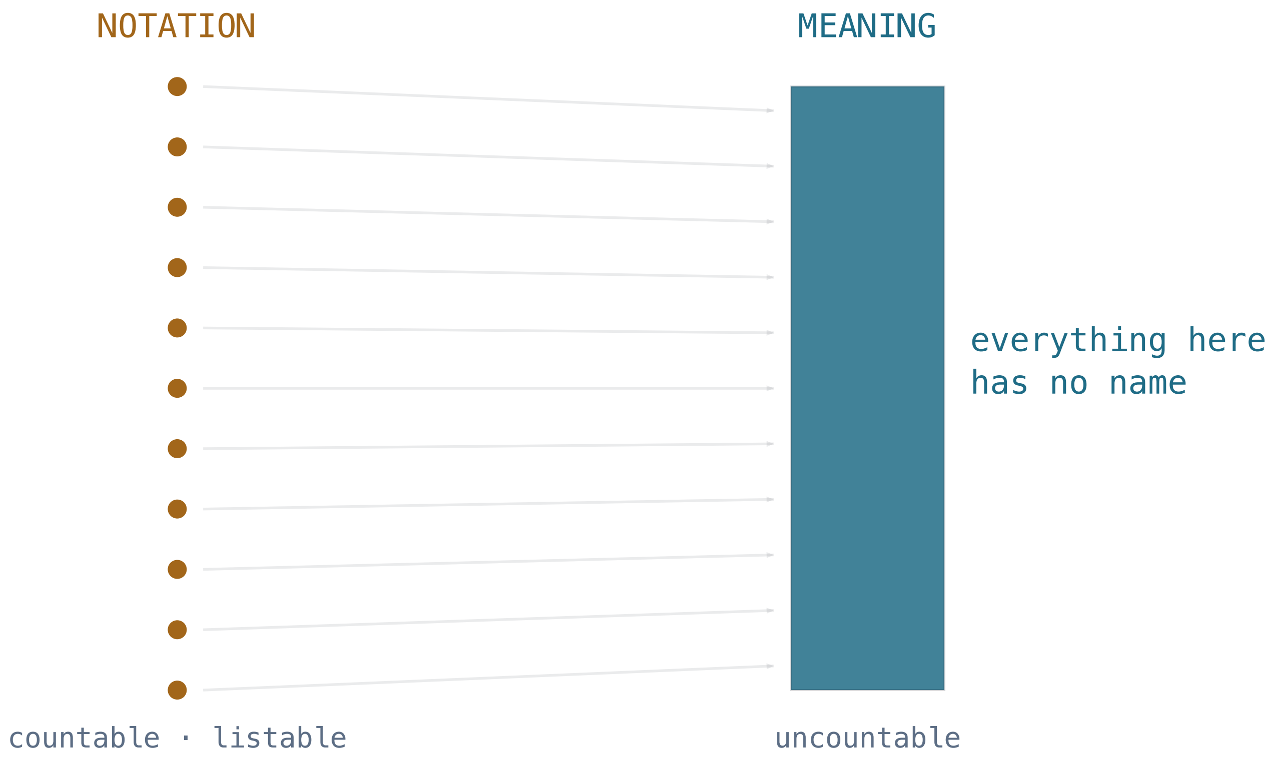
For every set S, the set of all subsets of S is strictly larger than S. In particular, the real numbers cannot be listed.

Where self-reference enters. The proof builds an object from the list that asks of each row: "do you contain yourself here?" — and then answers the opposite. The list is used against itself.

You have just seen it twice. Power set, real numbers, bit sequences: three costumes, one cardinality. Russell turned the same trick on set theory in 1901: the set of all sets that do not contain themselves.

THE RESULT EVERYTHING HANGS ON

There are incomparably more meanings than notations.



countable notation · uncountable meaning

Programs: countable. Behaviours: uncountable. So almost every behaviour has no program.

Proofs: countable. Truths about numbers: uncountable. So almost every truth has no proof.

Sentences: countable. Distinctions the world admits: uncountable. So almost everything is unsaid.

"Almost every" here is exact: the expressible is a vanishing sliver, of measure zero, inside the meaningful.

REFRAME · 2 OF 5

Scarcity of notation is a job description.

WHAT IT FORBIDS

A final language. No vocabulary — mathematical, legal, musical, or neural — will ever cover the space of meanings.

WHAT IT OPENS

An inexhaustible supply of things worth naming. Every new notation captures meaning that was previously unreachable — and there is always more left. Calculus, double-entry bookkeeping, staff notation, the periodic table, DNA sequencing, type systems: each was a raid on the uncountable.

Chaitin's version: almost all numbers are random, that is, incompressible, that is, nameless. Not a wall — unclaimed territory, and claiming it is what every discipline calls progress. This is the first time the title of the class shows through.

BEFORE NEXT WEEK

Recommended exercises.

1. Read the Meaning chapter from *What we want to talk about is not* to *Syntax and semantics*.
2. Make up six subsets of {1, …, 6} as answer sheets and build the diagonal D by hand. Check that it differs from every row.
3. Run the decimal version on a list of six decimals of your own. Then explain, in one sentence, why the fractions escape the argument.
4. How many programs of at most 1,000 characters are there over an alphabet of 100 symbols? How many behaviours on inputs 1 to 1,000 are there? Which number is bigger, and by how much?
5. Find one new notation from your own life — a recipe format, chess notation, a knitting pattern — and say what it made sayable.

NEXT WEEK

Self-reference I: a compiler defines the meaning of the language it is written in; the fixed point, and what Ken Thompson says it cannot prove; and Gödel's two theorems, with the same diagonal.