

REMARKS ON MATHEMATICS

24 NOV 2025

1. I'm inclined to adjust Curry's statement and understand mathematics as the *informal* science of formal systems.
2. When mathematicians prove that the subsets of N are uncountable with a diagonal argument, I'm now inclined to question whether subsets of N are sufficiently *defined* unless their characteristic function is computable.
3. My objection is related to constructivism, though I'd rather detail my concerns that defend this or that marginalism.
4. I'm not so concerned with feasibility, as finitism is understood to be, but I am against mathematical platonism.
5. Wildberger, a vivid online personality, argues his own critical case in terms of truth and reality. I am concerned, on the other hand, with making mathematical statements as exact and operational as possible.
6. I agree with intuitionism, more or less, that a timeless logic is not appropriate. If I have a computable characteristic function for $A \subset N$, then I don't already really "have" A but a process for *creating* A .
7. For instance, I don't think "all the primes" are already out there functioning as truth makers.
8. Of course I have a sense that feasible numbers can be checked with programs we'd all agree on. The stupid but obvious program is to try to divide n by all numbers m such that $1 < m < n$.
9. I suggest that numbers are something like equivalence classes of generalized numerals. We could use vertical strokes, with the successor function as the addition of a stroke on the right. So we generate N itself with $|, ||, |||, \dots$
10. So N itself is created in time by the activity of mathematicians. I speak of *generalized* numerals because we also speak of numbers,

“write” them with electricity, and so on. People say that numbers are types rather than tokens, and I’m adding that types are not immaterial entities in some platonic realm but our *enacting* the equivalence of tokens.

11. For context, I don’t think “truth” is a deep or meaningful concept. Instead we have situated belief and its transformation. In other words, belief is some organism’s belief. Scientific belief is belief that is relatively justified. Whether a particular belief is justified or not is another particular and always situated belief. But I believe that we do tend to share many beliefs. In practical terms, the demystification of truth is unimportant. But this demystification does affect my philosophy of mathematics.
12. Instead of “true but unprovable statements,” I prefer to speak of beliefs that we might want to be able to prove in a formal system. The formal “representative” of the informal belief is “meaningless” within the formal system, but our use of the formal system in the real world may involve a translation from the formal to the informal.
13. If a formal system coughs up P and $\neg P$ as theorems, then it might not work as a rule for action.
14. Personally I do tend to care whether formal systems “capture” something either about the practical world or about my mathematical intuition. I want math to be useful or beautiful or both.
15. My objection to platonism is “semantic.” I like the logical positivists, though I think they could be too credulous at times about the concept of truth. For instance, Ayer writes of “verification” while missing that his laudable phenomenalism shouldn’t include truth-makers. If verification is reframed as the temporary settling of situated belief, then I largely agree with Ayer. But using “verification” is misleading or incoherent.
16. One virtue of computable proof checkers is that some of the discourse of mathematics is normalized. Formal systems are intensely normal in the Kuhnian sense. But the science of formal systems is itself informal. Diagonal arguments tend to use proofs

by contradiction that seem to presuppose a background platonism.

17. Again, the power set of N is declared uncountable, but one could also question whether it remains meaningful in the first place, especially after informally grasping the countability of all programs.
18. One can prove (perhaps even formally, certainly informally) that the Lebesgue measure of the computable real numbers is 0. Indeed, the measure of any countable set is 0. But a study of computability theory might convince one that uncountable sets are not “really” defined. This would be an informal judgement, which might affect the way one evaluates the usual real number system with the usual classical logic.
19. One might even have a hunch that the system is without some hidden contradiction and still be disturbed by theorem that “most” real numbers are nameless and incompressible.
20. I recently saw this defended in terms of receiving bits random bits, but this real world scenario would only make sense if the number of bits were unbounded perhaps but always finite. Defining randomness is itself complex. We can understand it in terms of compressibility. But is this what we *informally* mean? A real world transmission of bits may or may not be substantially compressible. Perhaps we compute the expected information content of a random string of n bits. This turns out to be quite concrete, quite finite.
21. So this appeal to real-world sequences of bits that we receive passively and don’t determine by a rule doesn’t get us to those dark real numbers, which *can’t* be compressible, because then they’d be the output of a tape machine on some given input.
22. Dedekind’s cuts are insufficiently specified. The computable Dedekind cuts are of course countable. Because they are equivalent to computable non-decreasing characteristic functions on Q . We can encode the rationals in natural numbers, so we are talking about computable subsets of N again.

23. The bijection from Q to N is of course itself computable.
24. We want total functions, and a strong informal case has been made that we don't have a one-size-fits-all test for whether a program is a total function. Such a test would be something like our ability to formalize our study of formal systems. If we all automatically agreed, I suppose, that total computable functions were everything we should want.
25. Any computable function $f : N \rightarrow \{0, 1\}$ can be used to construct a countable list of functions of the same type in a computable way. We can then create a computable diagonalization that is not on the list. In fact, we can create a countable "set" of such quasi-diagonal functions. Or, more carefully, we can define a program for the creation of as many as we want and have space and time for.
26. I read this informally as an indication that any recipe for creating a countable list of computable bit sequences misses a sequence. I personally don't like framing this as a proof by contradiction. Because I don't accept Platonism.
27. Proofs by contradiction at the informal level — the level of actual human belief — seem to ask us to discover nonsense. But this means we are supposed to assume nonsense in order to see that it is nonsense.
28. I am also tempted to save the word "proof" for use only *within* formal systems. So a proof is a formal mathematical object. The science of formal systems, because it is informal, makes a case for beliefs *about* formal systems. Theorems are mathematical objects. I'd prefer to use some other word for the strong important beliefs that experts tend to share.
29. A mathematician may be happy enough working informally within a formal system. I say "informally" because typically only informal proofs are studied and written, with a sense that they *could* be formalized.
30. I loved real analysis, and I think epsilon-delta proofs give an informal insight that radiates beyond the formal system. In short,

some parts of real analysis are more convincing and “real” to me than others. The composition of continuous functions is continuous. “Of course.” I remain fascinated by attempts to capture the intuition of the continuum in discrete systems. Duals numbers are attractive in their simplicity. We can add the important transcendental functions. Or perhaps work with power series. “What is frankly hard for me to enjoy intuitively is equivalence classes of Cauchy sequences. These are too dynamic and too messy to “feel” like numerical analogues of points. Dedekind cuts, the computable kind, are better in this regard at least. But we still have equivalence classes of programs. We can transform back and forth, so here I am talking about an aesthetic issue. What construction “sings” to the intuition ? If any ?

31. With computable cuts, the order is not “already there.” Even with rational numbers it’s not generally *obvious* which is larger. But the computation is finite. Ignoring the issue of feasibility.
32. I confess a love for crystalline Q . The move to R is the move into partial functions and problems that can’t be decided in a bounded number of steps. This is real indeed if the reals are just the computable reals.
33. I credit finitism with caring about feasibility, but I don’t see a “natural” boundary. As computers we trust get more memory and faster, larger numbers become feasible.
34. Informally, we can come to trust larger calculations on larger numbers.
35. As formal proofs get bigger and bigger, we have to decide personally how much we trust a calculation. A platonist might call this issue merely epistemic. But for me numbers are necessarily embodied, even if (of course) no particular numeral is the number itself.
36. Belief is finally personal and more or less certain. As math is more formal and finite, consensus is more easily achieved. The vague belief by outsiders that mathematicians discover “absolute truth” is perhaps based on childhood experience with arithmetic,

which is of course an exposure to total computable functions. Applied of course to relatively tiny inputs.

37. “Calculation is an experiment.” A computable function, as deterministic, is ideally an extreme repeatable experiment. Even Platonistic functions are of course defined in terms of same input same output. Or more exactly as eternally frozen ordered pairs. The output is “already out there,” conjoined with input. The notion of computation as a process in time and material is stripped away. It’s as if all possible calculations happened before the world was created.
38. What contributes to the initial plausibility of this is perhaps the repeatability of a discovered algorithm. One learns how to enumerate the rational numbers. This trick is passed on. I say “discovered” when it was of course created. One imagines perhaps, implicitly, a time machine that one could use to carry the trick back thousands of years. It should work there too.
39. Some math feels so universal that one is indeed tempted to invoke eternity. It “could” have been created long before it in fact was created. Therefore (?) it was discovered rather than created.
40. Likewise the wheel “could” have been discovered before in fact it was. The wheel was hidden there in fallen tree, waiting to be sliced free.
41. I definitely enjoy the universality of math. I don’t want to evade the sense of discovery. And the creation of new math is constrained by what comes before. I told my student that “we make up rules and see what happens.” So in some contexts we experiment and discover the results of the system we put in place.
42. If we define a subset of N with a random tape machine, then we discover the results of that definition. But this discovery is active. So we create and discover at once. We run our program, that we created with coin flips, to see what it does. We can create something that only manifests itself in time.
43. This is the joy of the formalism. We have built a “machine” that can surprise us. We have subjected ourselves to a rule.

44. A program is a finite pattern that can generate “infinite” patterns. Patterns that have no built-in stopping point. We can put in more time and material and continue the “narrative.”
45. A program for the digits of $\sqrt{2}$ is inexhaustible. We break before it does. If we ignore feasibility. And, largely for aesthetic reasons, I think we should, at least where math is done just for the joy of it. In cryptography the feasibility is part of the game. I may construct a cipher as a work of art within the constraint of its security understood in terms of the size of the keyspace, etc. I could likewise study complexity theory for fun.