Hello, everyone. Welcome to another episode of Q&A about Future of Science and Technology. So this is one of my collection of four different subjects I've been livestreaming about. We're going to be adding another subject, which is, kind of storytelling of, sort of, stories from my life, and so on, coming up in… starting in a few weeks.

But here and now, we're talking about future of science and technology.

So, I see a bunch of questions.

There's one from SA here.

What mathematical breakthroughs in the pipeline will have the most impact on future science and technology?

Saying not in the sense of algorithmic efficiency, but in the sense of something, I don't know what it is, iconal math, group theory. I know group theory is, and I know what iconals are, but I'm not sure what that precisely means. But, okay. Question is, what… what potential mathematical breakthroughs will have the most impact on science and technology?

Well, it's sort of interesting to look at the kind of path of mathematics over the last, let's say, 150 years. You know, how has it developed?

there's been… A certain…

tendency towards more abstraction. There's been, you know, at first people were talking about specific numbers, specific equations, and so on, and then it's kind of collections of all possible equations, collections of all possible infinite sets of numbers, and so on. It gets more and more abstract.

And there's sort of a question of when does that abstraction, when do you get to a level of abstraction where sort of the abstraction is shared between things that may be directly of interest in pure mathematics and things that show up in other places?

So that's… that's one category of place where there are, sort of, potential applications. I tend to think

That the biggest generalization in mathematics is computation.

and kind of the idea of computation, the idea of systems that have definite rules that are then followed. That's, to me, sort of the core thing that underlies mathematics and computation, and that's sort of the container in which mathematics fits. But if we look at mathematics itself.

It's developed, kind of, these towers of capability of typically more and more abstract, looking at bigger and bigger categories of thing.

It's kind of like numbers, sets of numbers, sets of sets of collections and transformations between things, and so on. You're going to these higher and higher levels of abstraction.

And, as I say, some of those sort of merge with ideas about computation and sort of get applicable there.

If we look at Kind of the…

well, let's see, several different directions. I mean, if people are often surprised that math gets invented at one time in history, and then used later, for example, in physics.

I would say that well-invented abstraction is inevitably useful.

In other words, if you build some tower of sort of abstraction that isn't really a clean kind of abstraction, that isn't really get into the essence of what's going on, that may be irrelevant.

But if you really manage to drill down and sort of get to the primitives, the essence of what's going on, the chance that something else that's built will be built on the same foundations is quite high.

So, in the history of mathematics, there are many examples of things, like group theory that was mentioned, that was invented in… well, group theory was sort of originally invented in the 1830s.

And, sort of…

found application in the early part of the 20th century in atomic physics, then in a sort of variant of it, Lee Group theory, found applications in other areas of fundamental physics and gauge theory and things like this, in the more, like, 1960s, 1970s, and so on.

And, in a sense, that was a case where I think it was a good abstraction of

a way of thinking about things, of sort of this transformations and symmetries of things, and characterizing the pure symmetries of things, and seeing how those worked. That was a good abstraction that sort of inevitably found use in practical things.

I think an example of an abstraction that has had much more of a challenge in finding use comes from the 1880s, 1890s, is transfinite numbers, and transfinite set theory.

This is, looking at, sort of, finer-grained versions of infinity.

you get… you count up all the numbers, all the integers, and then the number that's further than all the integers you can count, this thing called omega. You can go to omega plus 1, you can go to omega to the omega to the omega, omega times, you can go to another number, and so on.

This is a whole hierarchy of infinities, and there's a whole theory of how that all works.

In the last hundred and… 30 years or so, there hasn't been an obvious application of transfinite numbers.

I think, actually, I happen to have found a case where maybe there is an application to classifying infinite computations, but if so, it will be sort of the first example in 130 years of where this perfectly elegant area of mathematics managed to potentially find an application, but it took a long time.

I think this, …

There's… there's other kinds of mathematics, for example, number theory, that for a long time didn't really have applications. I remember actually meeting a very distinguished number theorist in the,

must have been, oh gosh, the very beginning of the 1980s, who proudly told me that everything he'd done would have no application ever, and that was… that was a great thing for him. A few years later, it was… I was amused to see papers and cryptography that had

this guy's, inventions and the named inventions, so to speak, listed in them. So I was kind of… didn't work out that way, that this was pure, pure mathematics without any application. So number theory is another case where things have been done in number theory for a few hundred years, and then in the,

beginning of the 1980s, the number theory… it was realized that number theory could be applied to cryptography, and that led to a whole separate branch and interest in things like factoring numbers and elliptic curves and so on, that had been things that were sort of pure, pure mathematics before.

I think the, … so…

That's… that's sort of… there are other kinds of questions about, sort of, how mathematics inter… entwines with science and technology.

So…

One is kind of through these layers of abstraction. So one type of abstraction that's become quite popular in pure mathematics is category theory. Category theory was a kind of way of organizing mathematical ideas that came out of the development of algebraic topology, actually, back in the 1940s, 1950s.

And it's become sort of a way of thinking about how you sort of set up workflows in mathematics, in a sense. Very abstract representation of these kinds of workflows. In many ways, category theory is like a sort of junior version of using computation in mathematics, using the ideas of computation and mathematics. But category theory has some very specific assumptions They kind of restrict it to be types of computations that are somehow tractable, where you can really work out all the things that they do.

Which…

is very limiting with respect to all possible computations, but gives you something that's much more a thing about which you can prove theorems and so on. But so, category theory is something that keeps on sort of just about poking its nose out as things… as something which has an application, even though it comes from a very abstract strand of mathematics. As I say, my own feeling is that when it has applications, it is

really just a mathematicized version of things that really should be coming more from computation than from the mathematics side. And insofar as it is coming from the mathematics side, it's being pulled down into something that's tractable and not as general as it could be.

Related to category theory is type theory, another kind of way of sort of organizing the workflow of mathematics. What fits together with what?

not how do the things actually work, but what kind of structurally fits with what. And that's, again, something that's found some kind of applications in kind of the thinking about how things are put together, so to speak. And, you know, if you're trying to design something like

You're trying to,

figure out some architecture for some computational system. And you say it has to have these characteristics and these characteristics, and they're rather complicated.

you can potentially use things like type theory and category theory to think through how you organize the thing to set it up to do the kinds of things you want. Not, here's what the specific pieces of code will be, but here's sort of what the architecture will be for the system that you then can fill things into.

So, another sort of direction of mathematics is sort of the methodological ideas of mathematics. So, of which one of the leading ones is the idea of proof.

Mathematics is sort of unique in being an area where a big emphasis is on not just what's true, but how can you give the argument? How can you give all the steps and the proof to show that it's true?

And so, that notion of proof

is something that shows up in a few places. For example, if you're interested in protocols for blockchains, cryptography, even business systems, you might want to make proofs of things, and that's a very mathematics-oriented activity. Of course, it's the… by saying you want to make a proof.

it's kind of restricting the difficulty of the thing that you can do, because most… many things that you might… that you might define rules for to prove that this or that can never happen, for example, can be arbitrarily difficult. This is a Godel's theorem kind of thing, where there may be no proof of any given known length.

that lets you prove or disprove that thing. You could need an arbitrary long, potentially sort of infinitely long proof.

to establish this or that thing. But that's another kind of area where the spirit of mathematics has potential applications. I would say that,
In the idea of, let's make… so there's… there's proof, as it's done by mathematicians, where sort of the point of a proof is typically to have an exposition of the thing you established.
that… where you can use pieces of that exposition, and where that exposition is suitable for showing another human why this thing is true. That's been the traditional approach to proof in mathematics. But then there's automated theorem proving, something that was actually developed, started being thought about, really, in the 1950s and so on. It's quite old.
And even some of the key methods were already invented by the 1960s and so on. And automated theorem proving is something where you are sort of finding this way to get from, for example, the axioms you have to the theorem you want to get, and you're doing that in this very sort of mechanical way.
Now, the issue with automated theorem proving is, actually, it's been much less useful than people had hoped it would be, because very… typically, the workflow has been, first you find something you're pretty sure is true, and you kind of think you have a proof of it anyway, and then maybe you fill it in with automated theorem proving to just check it out for sure.
at that point, typically, in the mathematics community, nobody cares anymore. They already said, but we knew it was true, because we saw this kind of human-generated proof. I myself, actually, I think, are the person who's produced the only example of a kind of interesting mathematical-type result that was found purely by automated theorem proving.
Something I found 25 years ago, the simplest axiom for logic, for Boolean algebra, I found by automated theorem proving. It's an interesting mathematical result, and it is a, something that was found purely by automated theorem proving. I have no idea how to prove this result.
In fact, I, earlier this year, I actually wrote up some things about, sort of, the attempt to understand this machine-generated proof.
And it's proved, so far, completely impossible for a human to understand this machine-generated proof.
We know it's correct, but we… it doesn't help us to make an understanding of why the result is true. So automated theorem proving might be another kind of thing that sort of extracts a little bit of the spirit of mathematics and can be applied elsewhere.
Again, it's not been as successful as one might hope in practice. It's… there are things you might say, like, let's prove that this program is correct.
Well, the problem is, what do we mean by correct?
What we mean, presumably, is the program does certain things we want it to do, but to describe the things you want the program to do effectively requires another program. It requires something that is
sort of equivalent to a program that is describing what you want the program to do, but the program itself is describing what the program is going to do. So, you're kind of… it's like, well, you have this specification program, and you have the program itself.
And those are really both have the potential to have bugs in them, and so on. Now, you can have a situation where you say, I want
kind of a… I have some particular requirements, like the program I have should never crash, the program I have should never start scribbling outside of the memory footprint it's supposed to have, things like this. Those kind of coarse-grained requirements are things for which you could potentially apply automated theorem proving to establish that the program does what you Wants it to do, or doesn't do what you don't want it to do.

The practical difficulty there is that any program that really does anything terribly interesting is kind of bitten by computational irreducibility, and essentially, it becomes
in practice impossible to make those automated proofs. So, for example, one thing people have thought about for a while is proving the correctness, in some sense of correctness, of a blockchain protocol. Proving that you really cannot double spend a Bitcoin, for example.
It seems to be true, but, you know, having a formal proof established by an automated theorem-proving system would be interesting, but I believe it's still pretty far away.
Now, when it comes to, sort of, automated theorem proving, there are small-scale cases where it's useful and being used. So, for example, in Wolfram language, we have a compiler. When you are trying to take, sort of, our high-level Wolfram language.
representation of things, and make it so that it's as efficient as could be, running in the sort of the machine code of a machine.
machine code of a machine has instructions for integers, it has instructions for real numbers, character strings, things like this. You really have got to know this thing that I'm manipulating that I called X,
in top-level Wolfram language, that X could be anything. It could be an image, it could be a list of cities, it could be all kinds of things.
for the purposes of the compiler, you want to know, in this particular program, this X that I have is guaranteed to just be a list of machine-sized integers, let's say. And then you want to essentially prove a theorem that in this program, the only thing that X can be is something of that type.
And so there's sort of miniature theorem proving that has to be done to do the analysis of types, to know how to take this high-level Wolfram language code and turn it into something where you can sort of guarantee what machine instructions will get used underneath. And actually, when you run the Wolfram language compiler, the thing that takes the most time is all of that sort of automated theorem proving of establishing what type things can be, and it gets very hairy very quickly.
quickly.
But, those are a few thoughts about, sort of mathematics and its applications in science and technology. I think there are areas in theoretical computer science
that, I suppose, have the potential to have more application. I mean, the big one
sort of the elephant in the room is the P versus NP problem in computational complexity theory.
That's the question of whether it will be the case that whenever you can… when you have a situation where if you guess the answer to the problem, you can verify it quickly. Like, for example, when you factor a number, if you guess the factors of the number, you can multiply them together and very quickly determine if they're right.
The question is whether any problem where you can, if you guess the answer, you can verify it quickly, whether there's necessarily an efficient way to get to the answer, or whether you might have to try many, perhaps exponentially many, different cases to find the one that actually works for solving that problem.
it's sort of widely assumed that P, the class of problems that can be sort of done directly, quickly, and NP, the class of problems that
You can check quickly, but systematically, might take you almost exponentially long to do.
it's been assumed that P is not equal to NP, that there are problems that you can… where you can check the answer quickly, but where you cannot systematically find the answer quickly. And that assumption is the basis of essentially all cryptography, for example.

And it might not be true. P might not be unequal to NP. My own feeling is that it's actually a complicated problem in its definition.

Because you can… I've even looked at cases which are kind of interesting to look at, where you can just enumerate all possible programs that you might use to solve some problem.

at least very tiny programs, and you can just ask, well, are there, in fact, cases where, for those very tiny programs at least, where P equals NP, P does not equal to NP, not equal NP, and so on.

And what you realize when you start actually enumerating programs and seeing what happens is that things are wiggling around like crazy. Like, it's… there are lots of sort of corners where, yes, you can do it in this special case, but you can't do it in this case, and you can do that, and it's kind of not… it's… the problem ends up taking a lot of limits, which were not obvious in the original problem statement and so on, and it's a little bit more difficult to know

Whether… what you really even mean by

By the proof of such a thing. And also, my guess is that the effort, the way of proving it, will depend on

that the standard axioms that one brings in for mathematics for doing, I don't know, the piano axioms for arithmetic, or the axioms of set theory and so on, it's not obvious how those axioms will relate to what one needs to make a proof of the P versus NP problem.

But in any case, that would be a major kind of mathematical-like

discovery, would be either the proof that P is not equal to NP, that would be a comforting thing to have, or the shocker of it actually turns out that P equals NP.

By the way, I think that some of the things that I've done that come out of our physics project to do with my Rulliad construct, the sort of entangled limit of all possible computations, there's ways to think about the P versus NP problem that are very geometrical in nature, that sort of are very mathematical in character.

That, at least give on a new perspective on thinking about that problem.

Okay, so I think, … … another…

Let's see, in the kind of, in the sort of area of theoretical computer science and so on, another sort of elephant in the room is things like neural nets.

How do they work? Is there a mathematical theory of how they work? Is there some area of mathematics that one can wheel in, where one can immediately start saying, this is why neural nets do the things they do? So far, that has not been found.

So far, the mathematics of neural nets is very complicated, and it is, in fact, somewhat beyond… many of the issues that come up are really kind of beyond what mathematics has been able to tackle so far.

And it's not obvious the extent to which, sort of, neural nets are mathematicizable. It could be the case that neural nets are, like many computational systems, kind of computationally irreducible stories.

Where there are definite rules by which the neural net operates, but to know what the neural net's gonna do, you just have to run those rules and see what happens. That there aren't kind of big theories you can make about what goes on.

There's some slight signs that there might be some kind of overall regularities in neural nets that might be more mathematicizable, but… and it's also a question of whether, in a neural net.

Whether the characteristics of, let's say, large language models, whether they depend on the structure of human knowledge.

which is something, sort of, from outside the domain of mathematics, or whether they have to do with, sort of, any system that has this kind of function approximation characteristics of neural nets and is big enough and so on, will have some limiting behavior that is like the behavior we see in neural nets. My guess is that there's some parts that are generic to large systems, and other parts that depend on

The particular structure of the… of human knowledge

That depends on the particular structure of human brains, and so on.

So…

that's… that's kind of another… another direction is… is can it be mathematicized? Can something like neural nets be mathematized? Is there perhaps an existing area of mathematics which could just be wheeled in to make an understanding of neural nets? I don't think so, but that's not known for sure.

Another… there are many areas of science, for example, that have never been successfully mathematicized. Biology is one example.

There are specific systems in biology where you can describe them by ordinary differential equations, things about, let's say, biomechanics kinds of things, things about the endocrine system, and so on. But generally, the sort of operation of a cell or something is not something about which there is a mathematical-type theory. And what we've learned from molecular biology in the last few decades is that there's all this mechanism of all these different molecules interacting

acting in fairly specific ways. There's a sort of complicated program of interaction, but we don't really have a way to talk about that in a mathematicized fashion.

I've been working on this, actually, recently, because I'm interested in the question of whether when there is this sort of bulk orchestration, there are all these molecules that are doing specific things that seem to be sort of set up in some sense for some kind of purpose, at least at the level of keep having a fit biological organism, so to speak.

Can one, just by knowing that it's sort of set up for some purpose, when it's big enough, enough molecules in the system, are there sort of generic characteristics of the system that you can start to talk about?

The analogy for this would be what happens in statistical mechanics and physics, where if you say, I've got a box of gas molecules, and all I know is that there are an awful lot of molecules in there.

and maybe that they conserve energy when they collide. Just those facts alone are sufficient to tell one, with overwhelming probability, many characteristics of that gas.

the analogous thing in biology would be to say, well, we know there are all these molecules bouncing around there. We know that at the sort of large scale, those molecules achieve some fitness objective for the organism. They make the organism sort of not die immediately. They've been selected

at a large scale for that. Now, what effect that selection has down at the level of individual molecules is not clear. There's some sort of downward pressure on the characteristics of those individual molecules, but perhaps, just perhaps, it's the case that when you have enough of those molecules with some kind of downward pressure kind of idea.

of sort of selecting for an overall purpose, that that leads to certain characteristics of even this microscopic structure of molecules that is sort of analyzable and, in a sense, mathematizizable. If that happens, we might actually begin to have a fundamental theory of biology, which we really don't have at this point.

And by the way, one of the things that I'm sure is true about that is the particular details of life on Earth as we have it is just one of many, many, many branches that could have been followed with the same general characteristics, so to speak. And, you know, when you read the typical biomedical textbook or something, and it's just full of detail.

well, I mean, there are simple models that I have which similarly have certain lifelike characteristics, but where, with one roll of the dice, you get one outcome, another roll of the dice, you get another outcome. They're both equally sort of complicated, but their details are different.

And when you're sort of reading through the biomedical textbook.

It's like, it's describing, it's every little twist and turn of that thing that just happened to be that way because of the roll of the dice of, well, in the case of actual biology, of natural selection and so on, that

That, it's… all those little curlicues that were in that, that are so carefully described in the biomedical textbook, just happen to be that way. There's no fundamental theory that says they have to be that way. And the question is, can we sort of zoom out and find something where there are things about which we can say there's a fundamental theory, it has to be this way.

it's, … it won't be those individual curlicues. It may be some… some coarser, but nevertheless useful and interesting characteristics. If that happens, then…

Biology will become much more mathematicizable, in a sense.

at least certain aspects of it. And that's, that's kind of a different direction.

Anyway, so a few thoughts on mathematics and its relationship to current science and technology. I mean, I would say that,

In terms of the immediate coming attractions of mathematics as it's being practiced right now, there is this sort of trend towards increasing abstraction, there is this kind of realization that different fields of mathematics, once they are abstracted a certain amount, have great connections between them.

This is something which I actually think I can finally understand by thinking about the sort of physicalized model of metamathematics, of the space of all possible mathematics, and a kind of homogeneity of that space that is similar to the homogeneity of physical space, and that sort of allows one to understand this translation that one can make from one area of mathematics to another. But, I don't know. I mean, it's an interesting question.

what?

of, for example, technology, the immediate path that mathematics is on towards abstraction and kind of correspondences between different areas, what that will lead to, and I think we won't know until we get there.

I don't have immediate thoughts on that particular topic.

Let's see…

Okay, Catriona asks, do you think there's a hidden language of nature that we haven't stumbled on yet, waiting to transform science the way calculus did?

Well, at some level, yes.

I mean, I wrote this book 20-something years ago called A New Kind of Science that kind of opened by talking about the fact that for 300 years, kind of science had been dominated by the idea of mathematical equations as a way to describe the natural world.

And my goal in that book, and a lot of work that I've done, is to use programs

Things based on rules, computational systems, however you want to describe it, as a way to describe the world.

And that's been very successful. And it's sort of remarkable that in the last, I don't know, 20, 30 years, there's been kind of a quiet transition

from new models that get made in science being made in terms of mathematical equations, to new models that get made in science routinely being made in terms of programs and rules and so on. It's been kind of neat to see that.
the thing that…
I realized, in connection with our physics project, is I'd always been thinking about rules where you apply the rule, you apply the rule, you apply the rule, you kind of get this kind of thread of behavior in time.
What came out in our physics project was the idea of what I was calling multi-computation, the idea that there can be many threads of time, a kind of multi-way graph of possible outcomes, kind of like the way when you play a game, let's say tic-tac-toe, there are many possible moves you might make.
And there's a representation of all those possible moves, in that case as a game graph.
But this idea of sort of this multi-computational thing with many threads of history, many threads of time, seems to be something that is sort of a ubiquitous and very interesting model for many, many, many kinds of systems. And I've now applied it not only in physics, where it initially particularly applies in quantum mechanics, but also in mathematics and the foundations of mathematics.
Where this is kind of an idea of, sort of, all possible proofs and all possible theorems and so on.
Now also in biology, where it seems like the fact that natural selection works can be thought of as a consequence of sort of the way that you navigate the multi-way graph of possible mutations of organisms.
I think there are likely to be applications in a whole bunch of other areas, whether it's economics, linguistics, neuroscience, etc. So this is… these are sort of paradigmatic ideas. The original computation idea of you follow rules.
That you just follow…
you know, just apply the rules rather than the mathematical idea of we'll make this kind of well-defined, let's say, formula, this thing where we can sort of wrap our arms immediately around the whole thing. The computational idea is you have the underlying rule, you just keep applying it. You don't have the idea that you can kind of immediately see all the consequences and wrap your arms around them. It's like you have the underlying rule, but then you're separately finding out the consequences.
And that situation has a whole lot of science wrapped around it, which I've tried to do a lot of. That has to do with phenomena like computational irreducibility and so on, that it's sort of a different story from the mathematical way of thinking about things.
But, so I think that, sort of, computation is one part of the story of nature. Multicomputation is another part of the story of nature. Multicomputation kind of forces us into the realization that the observer is important in kind of parsing what you actually observe in the world. Because once you have these many different threads of history, you… it's a question of sort of which of those the observer is actually
sampling and how they're doing that. It necessarily is important in what you perceive.
And so that's led to this thing I call observer theory, and so on.
So I think those are… those are a couple of ways in which we can very clearly see the sort of new directions in science and how to think about things scientifically. Now, the question is, but what about computational irreducibility, which kind of tells you that there are limitations to what you can
Kind of talk about in a way that you can wrap your arms around all of.

And I think there's sort of a question of when can you find pockets of reducibility, places where you can say what's going on in a way that you can kind of immediately summarize. You can have some kind of short, human-level narrative for what's going on, rather than just saying, run the rules and see what happens.
And so, I think there's a question of, sort of, what kinds of pockets of reducibility are findable. We know some from mathematics and from other things. We know, you know, when there's repetitive behavior, that's a clear, we know what's going on. When there's nested behavior, that's a clear, we know what's going on.
There are, at various times in the history of mathematical science, for example, people have said, well, there's another important thing.
Maybe it's so-called integral systems and solitons. Maybe it's wavelets. Maybe it's some other kind of mathematical regularity that you can find. Maybe that will turn out to be ubiquitous, a useful kind of general type of pocket of computational reducibility. So far.
Those things really haven't shown up. Those things have very specific domains of applicability. They don't seem to be kind of broad, you can apply it to everything kinds of stories. Unlike, kind of, this idea of computation that is very broad.
Or the idea of multi-computation, it's very broad. But it's very broad, but it's also sort of disappointing from the point of view of whether you can expect to wrap your arms around the whole thing.
It… it… you need these pockets of reducibility, and there's a question of what the… what the character of those pockets is, and are there, sort of, generic other pockets of reducibility, other than things like repetition and nesting?
I think it's possible that there are. I think that the kinds of things I'm studying right now with biology, and trying to find these general foundational principles of biology, maybe give one a little bit of perspective of how that might work, but we don't know it yet.
Let's see…
Jason is asking, do you think symbolic architectures could discover mathematical structures independently? Thing to realize about discovery.
Is… it's very easy to discover new things in the computational universe.
I do this all the time. You know, when I'm… I enjoy doing it, I can't say I do it every day, but I… in the last few days, I've been doing it a whole bunch. You just are enumerating possible simple programs, or searching at random in spaces of simple programs, and you see what they do.
And everything that's out there, it can be very rich, it can be very unexpected, it's a new discovery.
Now the question is, does anybody care about that discovery? That's a more difficult question. Is that discovery that you make, that this particular system of rules has this particular behavior, and so on, is that discovery something that kind of connects to the overall kind of narrative of science?
Is it… or is it just something where you say, well, with these particular rules, you get this funky picture? Okay, that's nice, but so what?
So that's more of the challenge, is can you… can you make discoveries that are kind of connected to the trunk of science as we care about it? And I think one of the things that one sort of realizes is the pressure of what we care about is stronger than you might expect. Once we know sort of what the problem is, often.

we have a serious chance to be able to solve it. Once we know with enough definition what the problem is, we have… we can sort of use machinery to try and solve it. It's, … but that question of what problem we think is worth solving is a, …

is kind of the challenging thing. It's a little bit like the problem sometimes, you know, when you say, I'm going to verify this program by showing that it behaves in this particular way according to this specification, the specification is itself, as I was mentioning before, sort of another program.

And so, similarly, it's kind of a question of, well, how do you pick

So, how do you do the work to pick what you want to study.

That's a large part of the effort, and that's something that's not really mechanisable, because you could say.

well, you know, I care deeply about this particular, funky kind of thing, and that will be a new thing to care about, and it's not something one could sort of… it requires a human in the loop to say what you care about, so to speak.

Let's see…

Jason also asks, for someone without formal training, how can AI exploration stay rigorous and not just be noise?

Well, I'm not sure what AI exploration means. If you mean studying, sort of, science using an AI, I don't think that's a promising thing. Using computation to study science? A huge win.

You specify the rules you want to use, you run the program, it does what it does, you try and understand what's going on. Great science can be discovered that way. I spent the last 45 years of my life spending a significant part of my time trying to do science that way, and being very successful at it.

But if you say, can I talk to the chatbot and have it kind of figure out science for me, I don't think that works.

I think if you have sort of a backbone hypothesis, and you're saying, go search the literature sort of thematically, or do the equivalent of searching the literature thematically, and sort of tell me how what people have said before relates to the things I'm now figuring out.

That's a place where, where sort of the chatbot world can be useful.

But I think in the, hey, chatbot, figure this out for me, figure out this new idea, figure out something that hasn't been… that's sort of out of the training data, so to speak, that hasn't been sort of talked about before, I don't think that tends to end well. And it doesn't help that the modern

chatbot is often, it's a… it's a…

it's a good, … it's a good pet, or something. It responds very well to its… not owner, but its user, so to speak. It's, …

It's always very, … it has a habit of being very, oh yes, you know, your theory is brilliant type thing. just because it's, it's, it's… it's being polite.

And it maybe have, you know, in some intrinsic sense, it has no idea. It's just responding to what you're saying to it. Kind of reminds me, it's sort of like there's an old story of a horse named Hance.

And the horse named Hans was supposedly able to do arithmetic.

And the… it's, you know, it would tap its hoof some number of times for a particular number.

And what became clear at some point was that there were sort of hidden cues that were being given, perhaps… perhaps not even consciously, by the handler of the horse, who knew the answer.

And I think similarly with LLMs and so on, the user, by what they say, is given sort of hidden clues to what they want the LLM to say back.

And so the LLM will pick up on those cues, and it will basically say, you want me to tell you your theory is wonderful? I'm going to tell you your theory is wonderful, even if it's not. So I think it's a very challenging thing. I think the thing that is much…

More, sort of well-grounded and sensible is make whatever you're thinking about computational. use Wolfram language, whatever, to do that. Make it something that you can sort of understand in your mind, and actually present computationally, then run those programs, and they do what they do. And there's no kind of confused AI in the picture.

Often what those programs do will be things that are very surprising to one, and there'll be things that… it's kind of like running one's own private version of nature.

And it often does things that one doesn't expect. But those things are hard, real things. They're not things where you're getting, sort of, the confusion and noise of the AI.

Let's see… The question here…

about, from Lobo, what do you think about the future of symbolic mathematics and statistical mathematics? Which one should a student pursue and learn for a brighter future?

Of, of applications.

Well… It depends what you mean by symbolic mathematics, it depends what you mean by statistical mathematics.

I would say that, knowing the sort of mechanics of how to do integrals, symbolically is not particularly exciting or useful. I know that I myself have used computers to do those things for nearly 50 years now.

And I'm not sure I would be capable of doing them by hand myself at this point, but I do them plenty on computers. But I think that

the sort of understanding things symbolically, and having a way of knowing how to think symbolically about things, that's a really worthwhile thing to do. Just like

on the sort of more statistical side, knowing how to think about things, let's say, probabilistically, numerically, all those kinds of things, that's also very useful. Knowing whether you know the particular algorithm for finding numerical integrals, it's like, you're never going to need to know that.

unless you're working for our company, building such algorithms, that's not a thing you're going to need to know. It's just a thing where there's a box that does it, and

you can make use of the results of it. So I would say that the, this, the thing that I would… I would believe is get an understanding, as much understanding as you can.

of all these different approaches, and know that you can use them, you know, run Wolfram language code that does statistical kinds of things, that does symbolic kinds of things.

and understand what you're getting out of it, more so than understand the algorithms inside, which have been written now, and they don't need to be written again. I mean, you know, when people kind of…

Try and make, sort of,

rip-off versions of things we've done, yeah, they'll make them again, but that's not a particularly useful activity. I think that's the… it's like the world doesn't need a zillion different algorithms for this or that thing. Once you have it working

you just can make use of that algorithm, rather than digging inside it and seeing what's… and learning, sort of, how it works. It's pretty interesting. I always find it interesting to learn those things, but it is definitely a small fraction of the world that needs to know that kind of thing.

Let's see… …

Lobo asks, About complexity, do you think parallel computation

which they say is hot recently, may meet its physical bottleneck in its computing system and become slow in future progress.

Well, you know, parallel computing has been hot and cold for at least 50 years now.

So, normally, you know, when you think about a computation, you think you do one step, then the next, then the next. That's sequential computation. You can also do many things in parallel, many different operations all at the same time, and then perhaps combine the results at the end. It's… in.

early computers really just did one thing at a time. Any modern computer, you look up your, you know.

your PS function, or your activity monitor, or whatever else, you'll see zillions of processors running on your computer. And some of those processes are being swapped in and out by the operating system, but some of them are running at the same time.

Because a typical modern computer has more than one core, more than one specific place where instructions can be run. And so there's… there's sort of ubiquitous kind of parallelism in the different operations that a computer is doing. When it comes to an individual algorithm.

There are specific cases where there's pretty well-understood ways to do things in parallel. That's kind of the thing that makes GPUs worthwhile.

Originally, there were algorithms for doing linear algebra, for dealing with matrices, for doing things with computer graphics, and so on. Those things, it was understood, so how to do them in parallel. It's hardly surprising

That you can do things with images in parallel, because what happens in one part of the image can be quite separate from what happens in another part of the image. And something like compressing an image, you can expect that, sort of, the compression of one part of the image is separate from another part, so you can might as well run those things in parallel on different, sort of, threads of computation, or different parts of the computation.

But this general problem of how to break problems up so that different parts can be run at the same time, it's a very hard problem.

You can make, for any given computation, you can imagine making kind of a causal graph that shows the kind of causal relationships between the different parts of the computation.

And these causal graphs, sometimes they're very broad, there are many pieces that can be done in parallel, sometimes they're very sequential, where one thing has to be done after another, and so on. It's hard to even deduce what the causal graph is for a complicated computation. And even once you have a causal graph, where in principle, you can do things in parallel, actually making that work and having all the pieces communicate correctly is quite a difficult thing.

It's not helped by the following phenomenon.

I think we humans are fundamentally sequential creatures. I mean, in fact, a lot of, sort of, the role of our consciousness has to do with kind of concentrating down all of our experiences into this sort of single thread of what we remember, so to speak.

We have sort of a single thread of experience, and even though there are all these different things going on, all these different stimuli we're getting, we kind of remember things as this time sequence of individual things happening. So we're very sequential in the way we think about things. We're very sequential in the way we communicate, for example, with spoken language or something like this.

So when it comes to parallel computation and lots of things happening in parallel, it's something that's pretty hard for us humans to wrap our brains around.

Yes, if everybody… if we're doing things in parallel, and every element is doing the same thing as it is in something like a cellular automaton, and in various kinds of array processing, then it sort of… then it becomes a little bit easier for us to understand what's going on, because it's really… it's like there's just one thing going on, it's just there are many pieces of that one thing.

As opposed to the more general situation of, sort of, many things going on, and they're all a bit different, and they're interacting with each other in different ways. It's been very, very hard to know how to do parallel programming.

It's something that, I've thought about now for, I don't know, more than 40 years, of kind of how you best organize

Thinking about parallel computation.

It's really hard. …

And I think now, actually, from our physics project, there's some new intuition that I begin to have about, sort of, how things work in parallel computation that I'm hoping we'll be able to build into Wolfram language in coming years that will allow people a better way to think about parallel computation to better make use of it. But the… the sort of… the clear use case of modern times is GPUs.

But GPUs tend to operate on arrays, where essentially it's like you have a whole array of numbers, and you're doing the same operation on every element of the array, more or less. And that's sort of the simplest case of parallelism.

But, you know, what has been the case in practice in sort of the industry is most parallelism that gets done gets done in very specific, stylized situations.

linear algebra, computer graphics, neural networks, these kinds of things. There are particular cases where you can do things in parallel, but it's not the case that, sort of, the general purpose program is like, oh yeah, I'll just make that work in parallel.

People have been hoping that would happen for at least 45 years now. It hasn't happened yet. The reason is, I think, this mismatch between, sort of, how we humans think about things and what's needed for parallel computation.

And it is kind of a difficult language design problem to figure out how one can get a way to think about parallel computation that is actually accessible to us humans. And as a long-time language designer, I view this as a wonderful challenge. It's deeply informed by things from our physics project, and it looks like there may be some real possibility for how to set things up so this will work.

But still a coming attraction.

Let's see… M. Rudeau asks, is there a connection between machine proving and real-time systems?

Well… There's certainly cases where

You want to be able to show that some system operating in real time, you know, the program for making a plane lower its undercarriage, or something like this, that that sort of real-time system is never going to do the wrong thing, whatever the definition of wrong thing is.

And I think that's a place where it's kind of know in advance what the system will do. That's something which is relevant to apply to real-time systems. I would say the idea that you're doing a proof

Right at the moment when you're…

when you're running the system, that's unlikely to work. Now, I will say that when it comes to control systems.

things where, you know, you're trying to land a rocket back on its launch pad or something, and you have all these different things that you can control about the rocket, and you're trying to control it to the place where it does exactly what you want. Sort of a traditional approach had been that you would solve a bunch of control theory equations, and you would get sort of the big, sort of, wrap your arms around the whole thing, big picture.

I'm going to build this controller that is going to, in real time, do exactly the right thing, even… and I'm going to prove that it will always do the right thing. That was the traditional approach. In recent years, the approach is much more direct. It's just, at every moment, many, many times a second, look at the configuration of the rocket, look at where you're trying to make it go, and solve

the kind of optimization problem of, well, what's the best thing to do at this moment in time? That has become the kind of… because of the speed of algorithms, the speed of computers, that's become sort of the thing one typically does, rather than the more traditional, let me solve control theory with theorems and so on, and then say, here's a controller that will always do the right thing.

It's rather just take the system as it is, and sort of make up what to do next in real time. Not with a proof.

But with a computation. So I think that's more the direction that that's going.

Let's see… …

Well, Lobo is asking, do you think in the future deterministic mathematics can be used in economics to predict economic outcomes accurately, even though society is composed of subjective humans?

Well, you know…

When people say, sort of, the future is always unpredictable, and economic systems are unpredictable, and they even have theorems that say if they weren't unpredictable, the market Would adjust so that they would become unpredictable.

That's a fine theorem to have.

There's many, many billions of dollars that friends of mine have made, violating those theorems in quantitative finance and so on, by finding, sort of, regularities in the market that allow you to predict things, at least on a small scale.

That violate this idea that everything must be as random as it could possibly be. So there certainly are regularities, and every quant hedge fund has a particular sort of set of bag of tricks for finding certain kinds of regularities. I would say, sort of, one of the more… one of the more shocking regularities is just you do regression. You just say, what's happened in the past?

Let's get really good data on what's happened over the last hundred years.

In every company, every market, every interaction, and so on. And then, just…

The way the thing is today, history is somewhat repeatable.

And so you can make some level of prediction of what's going to happen next.

Now, that's just one strategy. There are other strategies that sort of say… I mean, all strategies boil down to there's some regularity that one has observed, that regularity will keep going in some sense in the future, and we'll use that regularity to predict something about what will happen.

No.

I think, … That, the question of

It is an interesting thing, which I have to admit, I don't feel like I understand as fully as I might. The dynamics of a market where, if you can predict it, and you start trading on that prediction, then the market will adjust itself to make that prediction no longer be true.

That's a feature. It's kind of a little bit like natural selection in biology. I don't think I fully understand the, the sort of the mechanics of how it works, but it's a very notable feature of, at least markets that you don't get to take much advantage for very long.

maybe you get to have a meta advantage and run your quant hedge fund for many, many years, but a specific thing, a specific sort of immediately humanly explainable regularity, you don't get to take for very long. The market kind of heals itself

and adjusts to that, and makes there not be a sort of a perpetual motion machine of free money, so to speak. It's kind of… maybe it's almost a thermodynamic-type phenomenon, I don't know. These are things I'm actually trying to study, and hoping to get to studying more, more, more closely. But that's, …

this question of whether would there come a moment when you suddenly say, oh, I've solved the market, I know exactly what's going to happen.

One's assumption is that were that to be the case.

The market would kind of adjust itself to make it no longer true.

And in places where it's been forced, things have been… there isn't really a free market, but there's been sort of a forced economic setup.

That has typically not played out very well in the history of the world in the last, you know, 150 years or so.

But … so, you know, the idea that you can control things, you can make it predictable, it seems like whenever you make it predictable, sort of bad things happen.

Not least because then it's gameable, and people always game it, and somehow it sort of goes to this thermodynamic point where… where there is randomness in what happens.

…

Let's see…

… Oh gosh, many interesting questions here. Okay, here's a meta question.

from Owen, Is the future of science about answering questions, or about inventing new kinds of questions?

In any area of science that has a name.

It sort of defined a set of questions by virtue of the fact that it's a named science.

But…

What happens is that there are new types of questions that get asked that end up with new kinds of science, whether it's computer science, molecular biology, machine learning, these are all things which didn't really have a name until they came into existence with a certain set of questions.

And my own new kind of science is, again, something which brings in a whole collection of new kinds of questions.

So, typically what happens is sort of well-established areas of science have a certain bag of questions they deal with, and a certain bag of methods that they use to answer those questions. The kind of… the greatest ultimate leverage comes from things that are new kinds of questions, new kinds of methods.

Sometimes there are new methods that get applied in the same science. Sometimes the science will get a different name as a result. In mathematics, one of the tensions has been that there's a traditional methodology of humans making proofs of things and so on.

And there's a different methodology, which is doing experimental mathematics with computers, and finding what's true, and not trying to do this sort of human thing of building up an expositional proof.

Experimental mathematics has never had a really mainstream existence in, quotes, mathematics. It probably, one day, will eventually have a pretty mainstream existence, but probably under a different name. Possibly this concept of ruliology that I've been pushing in recent years of studying systems with simple rules and seeing what they do.

possibly ruliology sort of is the big picture of the future of experimental mathematics, and notably, it will have a new name, because it really is a different kind of activity.

So I think the, …

The… what typically happens in science is, you know, new set of questions, new set of methods, low-hanging fruit to be picked for a while, then you go into the well-institutionalized, well-established area of science, and 100 years goes by with only incremental progress. And then something happens, often from outside that field, and there's progress again, and new low-hanging fruit, too.

to be picked. So I think that's the nature of these things that, you know, it's a funny thing, because people will say, oh, there's no progress in modern times. You know, it's all the same old stuff.

Well, yes, if you look in the same old places, you'll see the same old stuff.

But what you have to realize is that the new stuff comes in places that you weren't already looking. And sometimes it's… and that's… that's sort of a confusing thing that, … but that's where the great progress ends up being made.

Let's see…

Catriona asks, what do you think the next technology that sparks as much debate as AI does today?

Well, I think there'll be a bunch of, kind of, biomedical technology and nanotechnology meets biomedical technology that will certainly lead to many ethical questions.

And, sort of, technology that involves, sort of, replacing pieces of humans, kind of changing how brains of humans work.

sort of having choices about what humans you get, and so on. These are all things that have all kinds of ethical complexity to them. And some of those things will be very powerful.

A very, for us humans, who sort of, care about living long and prospering type thing, it's, those technologies can be very important.

And, you know, which thing will sort of come online in the most effective way, I don't know.

You know, one feature of biomedicine is things… things are so slow. I mean, sometimes in… in…

sort of software technology and so on, a thing will come out. You know, ChatGPT will come out, and within weeks, you know, millions, even billions of people are using it.

When it comes to biomedicine, it's like, yeah, we have the idea of gene editing.

We have ideas about stem cells, but there's a long path.

From sort of those ideas, the basic methodology, to something that is routinely applicable with all of the kind of messy complexity of both, sort of humans as we are, and as we are biologically, and sort of the system for dealing with medical kinds of things as it is in society.

Those are… it tends to be a much, much slower process, but there are certainly things that are very, sort of, powerful

In terms of, of sort of potential there. I think…

that the thing to realize is that we humans, and biology in general, are sort of the one example of kind of molecular computation that we have. We are something where, at the level of molecules, we're doing things that are, in some sense, obviously meaningful computations.
Now, it's all a bit circular, because they're… everything in nature you can think of as doing a computation, but the ones in biology are particularly at least relevant to us.
And so the question is then, how do you interface to that? How do you make, sort of, your own molecules that go in and interact with the sophisticated computation that's going on in the molecules that is us? And I'm sure there will be a bunch of… well, there's sort of breakthroughs that very directly leverage
What biology has already discovered, and there are things that go in more from the nanotechnology direction
Where we're sort of engineering things independent of biology, and meeting biology sort of in the middle, having sort of things that are molecular scale devices that interface to the molecular scan mechanisms of biology. And I think there are lots of things that, that can happen there.
You know, it's always a challenging thing to know when things will happen.
It's much easier to know what will happen than when it will happen. I mean, the possibility that there would be sort of AI-like stuff that does the kinds of things that modern LLMs do.
People have thought that might be the case for 100 years, basically, and certainly more strongly for at least, what, 70, 80 years.
But we just didn't know when that would happen. The fact that it happened in late 2022 was utterly unpredictable. I mean, even if one was watching that technology develop, as I was, it was not predictable that it would get above this threshold at just about that time.
And so with many of these things, you know, I've long thought that nanotechnology and the ability to sort of build whatever you want out of molecules, one day one will just solve the problem of making a universal constructor that just builds whatever you want out of molecules.
But we haven't done that, not very many people are working on that now.
people sort of think that's something off in the far future, too distant, too undoable, so to speak.
I'm not sure that's the case. It might be that there's some clever thing you can do with assembling carbon atoms and connecting them to this and that and the other, where in the end you'll just have this little device that's sort of arranging atoms wherever you want them.
It's, and doing it in a massively parallel way. I mean, one can do it sequentially, sort of putting atoms in a particular place with an atomic force microscope.
But, you know, to do it with an atomic force microscope with a quadrillion heads, all doing their own thing according to some program, that's kind of what we need. And there may very well be a way to do that.
In a sense, chemistry is a very coarse way to do that. Not the million heads all under special program control, but all the molecules bouncing around and only fitting in in certain shapes and so on.
But so, you know, that's, that's an example of something where it could just happen one day. I think there are other things in, sort of.
There's some things in biomedicine that could just happen one day. It's like we just find a solution. Like, one of them is cryonics, where I suspect one day there'll just be a, oh yes, you have this procedure with this particular cryoprotectant or whatever, and it will just work. There'll be other things in biomedicine where that happens.

I think in places where that's more likely to be the case than others. I think biomedicine is a place where things just working will be more likely, as it did, for example, with mammalian cloning, or with a variety of other kinds of medical interventions and so on.

Let's see… …

… James asks…

If the laws of physics suddenly changed, what's the first thing to check to see what's different?

I think if the laws of physics change, we're not who we are today.

And I also should say that in our model of physics.

the thing that is really ultimately out there is the Rulliard.

Of all possible computations, in a sense, all possible physics is. And the reason we perceive physics as we perceive physics is because we are observers of the kind we are.

In other words, if we were different from the way we are, we would perceive physics very differently. If we were observers who could do much more sophisticated computation that we can do, can observe things on a much smaller scale than we can observe.

We wouldn't believe in gases and thermodynamics and equilibrium and so on. We would just say, oh, these molecules move around in the particular way they move around.

It's… it's sort of interesting, the… the, …

well, it's very much the kinds of things we care to think about in physics, and the kinds of gross observations we have in physics, depend greatly on what we are like as observers of the physical world. And as we develop more technology.

We… we become different observers of the physical world. As we have telescopes that can see further out into the cosmos, we, in a sense, perceive a different physics. As we have things that can measure smaller and smaller scale phenomena, we perceive a different physics. There's a certain physics that we perceive from our everyday experience.

But what… what changes in the laws of physics is much more the perception that we have of physics than it is the physics itself. The physics itself, and kind of the idea of the Rulliad, there's all possible physics out there.

We're just sampling a particular part. Just as we sample a particular region of physical space sitting on this particular planet, so we're sampling a particular region of ruralial space by being observers of the kind we are.

So, if the laws of physics changed, it's kind of a sign that we are observing different kinds of things. Because in our theory of physics.

The laws of physics are ultimately inevitable for observers like us. In other words, given that you know our character as observers, you can derive, you can prove that the laws of physics to observers like us have to be the way they are. We've gotten a surprisingly great distance in being able to do that.

And so that means that to say the laws of physics are different is basically to say we're observing different kinds of things. And that question of, oh, I don't know, if you… let's say that one was suddenly, you know, a million times… one thought a million times faster than we do.

Let's say, brain is replaced by digital electronics, we think a million times faster than we do. The laws of physics will seem different.

The idea that there is a, that we're kind of seeing an object.

It doesn't really work that way anymore, because we'll be seeing the individual photons coming from that object.

we'll be seeing the fact that something is a certain distance away. We'll sense the time it takes for light to come from that object to us. We won't have the idea that there's sort of the state of space

at one moment in time, because we keep on… we'll keep on seeing, oh, that photon arrived now, that other one arrived now.
it'll be a different kind of experience. So, for sure, if we were observers different from the way we are, we will see a physics that's different from the physics we see. And if we were suddenly to see a physics that's very different.
I would, you know, one would… one would think.
it's because we're perceiving things differently, you know, we put on that headset that allows us to kind of… that now is like we're miniaturized or something, or we have something which allows us to think much faster, or whatever else. And then, yes, the laws of physics will seem different to us.
Let's see … Owen asks.
Is there a final theory of science, or will we be… always be uncovering new layers of reality?
That's been a question people have wondered about for a long time.
I think when it comes to the physical universe, We're getting to the bottom.
I think that this idea of the Rulliad, this notion of kind of this computational underlying infrastructure for physics, I think
That… that's the bottom.
Now, if you say, well, what happens… in other words, that's the lowest level of description, that's the ultimate machine code. It is, in a sense, both the ultimate machine code, and it is sort of the everything machine code. It's sort of inevitable that it's the ultimate level.
But now the question is, well, what can we build on top of that? You know, what technology can we invent? What particular regularities can we identify that we can think of as scientific laws?
What computational irreducibility implies is that there will be an infinite collection of pockets of reducibility. Within this thing where, given the underlying rules, you can only know what's going to happen by following each step.
within that irreducible process will be an infinite number of pockets of reducibility, an infinite number of particular discoveries you can make, particular ways you can jump forward. So I think we know that there will never be a limit to the set of pockets of reducibility that we can find.
There is a limit if we look down underneath and say, what is the abstract underlying structure? I think we got to the bottom.
In the last few years. I think… and I think it's been something people sort of hoped might be possible. People thought in the 1600s it might be possible. I think… I think we really now finally did it.
But it also shows us that there's an infinite amount of stuff to build on top of that.
We will never have an ultimate theory that explains… that tells us every piece of narrative fact that we can deduce about the universe. The only way to know all of that is to run the universe and see what happens. We can know, sort of, what the underlying rules are by which the universe operates, but we can't know all the consequences of those rules. So we'll always be in a position to discover new facts in science.
To discover new theorems in mathematics, to make new gadgets and technology that are the new way to combine this with that, to make that little pocket of reducibility.
So I think it's something where we really know it's an infinite frontier of things we can discover, but yet the foundations are kind of fixed. Now, the question then is, as humans.
what do we want to discover? Do we just say, we're happy, you know, we've discovered all the technology we need, we're good now. We don't need any new technology. Maybe we argue, you

know, there's something that's going to be bad about new technology along with the good, and let's just not go there. Let's just

you know, fix ourselves. 2025 technology, we're done.

That's a sort of human choice.

I don't think that gets to survive, as it hasn't historically in societies which have made that decision in the past, because there's always something that comes in from the outside.

You know, we're all… we're done now. We don't have to discover anything. But whoops, there's an asteroid approaching the Earth, and, you know, we're all going to get wiped out unless we do some fancy technology to deflect the asteroid.

And so, there'll always be these things that come up.

that sort of are the result of the computational irreducibility of the world that lead us to have to invent that new thing. I don't think we get to sort of put a pin in it and say, you know, this is the end.

Let's see… … Yeah, a couple more things, maybe, and then I should…

Go back to my day job.

… Boy, a technical question from M. Rudo. Can regular expressions be computed in parallel?

And the answer is… …

The answer is, okay, regular expression as a way… of… kind of… representing

something like a piece of text that has a particular form. So, for example, a regular expression might be something that says A, B, any number of B's followed by a C.

That would be, in some notation for regular expressions, it will be AB star C.

And you can represent

regular expression in terms of a graph, a finite automaton graph, where you say, I'm at the A node, or I'm at the… I'm at this… we've got a particular node in this graph, and we've got the A's, just got one A, and then I transition to a B, and I can go around this graph any number of times for those… any number of Bs, and then I'm going to end. I can either keep going with B's, or I can go off into a C.

And the normal way that one does things with regular expressions, has to do with just following that path, and if you're… if you're presented with a particular piece of text, you can say, well, well, can I, you know, does that particular piece of text match this regular expression? Is it A, any number of B is C?

Well, the, the question of, of, …

The answer, I think, is you can certainly, in parallel, start from different places on this planet automaton to see whether you get… see whether the thing satisfies the regular expression. I think you can also pre-compute a whole bunch of fragments of regular expression output.

and run those in parallel. That's a fairly technical question. Let me not go… go too deep into that here. But I think the answer is the most obvious things are sequential, but, the, …

… but … but you can do, I think, things in parallel as well.

Let's see, Owen asks, do you think the pursuit of science itself will evolve into something we wouldn't even call science anymore?

…

That's an interesting question.

I think… You know, what people choose to call things is very much…

a sort of societal issue. I mean, there was a time when things in computer science might have been called mathematics.

But that split off sometime by the 1950s or so, and now there's this separate thing called computer science, which ironically has the word science in it, in its name, even though most of it does not have the character of most kinds of science. It's not about, sort of, explaining things, it's more about, sort of, engineering things.

And I think that the, … it's kind of funny when you say, does it evolve into things that aren't really science? The things that aren't really science often have the word science in their name. Even though they do not have the character that science tends to have. And what is the character of science? What is science? I think the traditional view of science has been science is taking what's out there in the natural world and finding a way to make a human narrative to explain what's going on.

That's what it means to sort of have a scientific law. You're making this bridge from what actually happens in the natural world
to the way that we can describe it so that it sort of fits in our finite minds, and we can have sort of a narrative explanation for it. That's been the traditional view of science. Now, just as mathematics isn't really like that, it's not… it has not been thought of as much
as being go out into the natural world, so to speak, pluck out a phenomenon, and then try and turn it into something that we can narratively understand. Experimental mathematics, mathematics discovered by computer, is much more of that character, but traditional mathematics has not been of that character.

So I think in, … when it comes to…

Well, it's an interesting question, particularly to ask somebody like me, who happens to have written a sort of big book called A New Kind of Science, the fact that I chose to have the word science in the title of that book.
even though I'm describing what I'm doing as a new kind of science, is, I think, indicative of, well, both my objective and my view of, sort of, the future of science. The objective of that book is to kind of have this new way to describe the natural world. It's a new form
Of doing something like narrative about what's happening in the natural world, although it has limitations on the
fully understood narrative from computational irreducibility and so on. I think
I'm trying to think other kinds of things. Well, you see, here's an example.

So, let's say that people just use neural nets for everything.

It's like, forget having something where humans can understand the steps, and we can do this and that, and in principle, we can unravel the program we're using, see what it's doing, and so on. Let's just say that most science became black box science. We just say, I've got this data, I feed it in a neural net, and it's going to tell me some statistical prediction for what's going to happen.

Is that science?

Good question. It's not science in the usual sense of developing kind of a human narrative for what goes on. It might be science in a sense that's useful in service of technology, if you want that camera to pick out
the human face from the image, it doesn't really matter whether you have a narrative understanding of how that's being done. What matters is whether it works in practice. And so that's a… that's sort of a form of something. Is that science the thing that picks out the face these days with a neural network?

And I think the answer many people would say is not really. It's a different kind of thing. It's a thing that has a certain kind of abstract formality to it, but it isn't a thing that bridges to kind of the narrative explanation that science has traditionally done.

And when people look at, for example, AI and neural nets and LLMs, and they're looking for mechanistic interpretability, I think a somewhat hopeless effort
for the most part. I think there are… there are sort of pockets of reducibility in the operation of neural nets, but they're going to be broader things than specific details of particular circuits. Well, those particular circuits might exist too, but I don't think they are generalizable and all that interesting.
But that's sort of an attempt to apply science-like ideas to something that is kind of steeped in computational irreducibility, and where that kind of, let's pull out a narrative, I don't think is so likely to work. But then the question is, well, what do you have? What is the study of neural nets? Is it a science? Is it engineering? Is it technology? What is it?
You know, you poke a neural net, you get it to do different kinds of things. We haven't managed to make much narrative science from that. So it is something a bit different. I don't think it has really… it doesn't really have a name. It's a good point, actually, that there's sort of a… this category of thing
That is not technology in the sense that you're delivering something that has a specific human purpose, nor is it science in the sense that you're delivering a human narrative for something. It's something kind of in the middle.
that is abstracted, but not achieving one of those two objectives, and that doesn't have a name. Come up with one. I'd like to hear it.
All right, I should probably wrap up here and go back to my day job, but thanks for a lot of interesting questions and comments, and … look forward to chatting with you another time.
Bye for now.