The following is a conversation with Stephen Wolfram,
his fourth time on this podcast.
He's a computer scientist, mathematician,
theoretical physicist, and the founder of Wolfram Research,
a company behind Mathematica, Wolfram Alpha,
Wolfram Language, and the Wolfram Physics
and Metamathematics projects.
He has been a pioneer in exploring
the computational nature of reality,
and so he's the perfect person to explore with together
the new, quickly evolving landscape
of large language models as human civilization journeys
towards building super-intelligent AGI.
And now a quick few second mention of each sponsor.
Check them out in the description.
It's the best way to support this podcast.
We got masterclass for learning better help
for mental health and insight tracker
for tracking your biological data.
Choose wisely, my friends.
Also, if you want to work with our amazing team,
we're always hiring, go to lexfreedmen.com slash hiring.
And now onto the full ad reads.
As always, no ads in the middle.
I try to make this interesting,
but if you must skip them, friends,
please still check out the sponsors.
I enjoy their stuff, maybe you will too.
This show is brought to you by masterclass.
180 bucks a year gets you an all-access pass
to watch courses from the best people in the world
in their respective disciplines.
There's several components to effective learning.
I think learning the foundations is really important
and the best way to do that, depending on the field,
is probably some kind of material
that encapsulates the foundations.
It could be textbook, it could be a really good YouTube video,
a really good tutorial, whether written or video form.
Then there's the actual practice of those foundations
by building something.
Again, it depends on the field.
But I think a component of learning

that's often not utilized is to learn
from the best people in the world
that did that thing you're trying to learn.
I think even if they don't cover
the entirety of the foundations,
even if they don't cover a kind of hands-on tutorial
type of description that you can get elsewhere,
through their words, you can get the wisdom of the details
that mastery in that field requires.
And you could also see kind of take in the mode of being
required to achieve mastery in that field.
I think it's so powerful that Masterclass allows you
to look in to some of these world experts
in a structured context, really intensely learn from them,
not just the content, but the way of being.
I've listened to so many of them,
it's too long to list.
But Carlos Santana will write Daniel Negrano
before I did the podcast with him.
These are all just really excellent, Gene Goodall.
If you wanna check it out,
go to masterclass.com slash lex
to get up to 35% off from Mother's Day.
That's masterclass.com slash lex for up to 35% off.
This episode is also brought to you by BetterHelp,
spelled H-E-L-P, help.
I posted this meme on Twitter recently
that has that meme format where the car swars off
on an exit and going straight means going to a therapist
and swarming off onto an exit says,
saying in quotes, it is what it is.
And then the car is labeled as most men.
It's true.
I think a lot of us face hardship in life.
And I think there's a dance between kind of
being fragile to the richness of the experience
of that hardship can really break you.
So there's some usefulness that it is what it is.
But afterwards or during it,
there has to be some component where you're raw
and honest with your feelings
and you bring them to the surface with yourself
and you introspect what you think,

what you feel, what you fear, what you hope.
It is so simple, but so many of us are afraid
of the simplicity of that intense feeling
that our mind is capable of,
that roller coaster that our mind takes us on.
So I think therapy, bringing stuff to the surface
with a licensed professional
is definitely something I recommend.
Mental health is at the core
of what it means to be a healthy human being.
And BetterHelp is easy, discreet, affordable,
and it's available everywhere.
Check them out at betterhelp.com slash lex
and save on your first month.
That's betterhelp.com slash lex.
This show is also brought to you by Insight Tracker,
a service I used to track biological data,
markers from my biology, from the blood tests they take.
It looks at blood data, DNA data, fitness tracker data,
all that kind of data coming from my body
to help me make decisions about my lifestyle.
The more conversations I've had with biologists,
computational biologists, biochemists, bioengineers,
neurobiologists, so people specialize
on particular systems within the body,
virologists, immunologists, all of that.
I realize how incredibly human body is,
how incredibly machinery of it is,
and how many signals it provides internally
for that large-scale, hierarchical system
to maintain equilibrium, to maintain health,
to maintain life in the full definition of those words.
And I think it's a really exciting possibility
in the future that we can get
as much of that signal as possible.
Richly, temporal signal, every second of every moment
from every system within the body
and help us make predictions about where stuff goes wrong,
helps, gives us advice on what we should do.
And so I think services like Inside Tracker
is a really important step into that direction.
Get special savings for a limited time
when you go to insidetracker.com slash lex.

This is the Lex Friedman podcast.
To support it, please check out our sponsors
in the description.
And now, dear friends, here's Steven Wolfram.
["Inside Tracker"]
You've announced the integration of chat GPT
and Wolfram Alpha and Wolfram Language.
So let's talk about that integration.
What are the key differences?
From the high philosophical level,
maybe the technical level between the capabilities
of broadly speaking the two kinds of systems,
large language models,
and this computational, gigantic computational system
infrastructure that is Wolfram Alpha.
Yeah, so what does something like chat GPT do?
It's mostly focused on make language like the language
that humans have made and put on the web and so on.
So it's primary sort of underlying technical thing is
you've given a prompt, it's trying to continue that prompt
in a way that's somehow typical of what it's seen
based on a trillion words of text
that humans have written on the web.
And the way it's doing that is with something
which is probably quite similar to the way we humans
do the first stages of that using a neural net and so on.
And just saying, given this piece of text,
let's ripple through the neural net
and get one word at a time of output.
And it's kind of a shallow computation
on a large amount of kind of training data
that is what we humans have put on the web.
That's a different thing from sort of the computational stack
that I spent the last, I don't know, 40 years or so building
which has to do with what can you compute many steps,
potentially a very deep computation.
It's not sort of taking the statistics
of what we humans have produced
and trying to continue things based on that statistics.
Instead, it's trying to take kind of the formal structure
that we've created in our civilization,
whether it's from mathematics
or whether it's from kind of systematic knowledge

of all kinds and use that to do arbitrarily deep computations
to figure out things that aren't just let's match
what's already been kind of said on the web,
but let's potentially be able to compute something new
and different that's never been computed before.
So as a practical matter,
our goal is to have made as much as possible
of the world computable in the sense that
if there's a question that in principle is answerable
from some sort of expert knowledge that's been accumulated,
we can compute the answer to that question
and we can do it in a sort of reliable way
that's the best one can do given what the expertise
that our civilization has accumulated.
It's a much more sort of labor intensive
on the side of kind of being creating
kind of the computational system to do that.
Obviously, in the kind of the chat GPT world,
it's like take things which were produced
for quite other purposes,
namely all the things we've written out on the web and so on
and sort of forage from that things
which are like what's been written on the web.
So I think as a practical point of view,
I view sort of the chat GPT thing as being wide and shallow
and what we're trying to do with sort of building out
computation as being this sort of deep, also broad,
but most importantly kind of deep type of thing.
I think another way to think about this is
if you go back in human history,
I don't know, 1,000 years or something
and you say what can the typical person,
what's the typical person going to figure out?
Well, the answer is there are certain kinds of things
that we humans can quickly figure out.
That's sort of what our neural architecture
and the kinds of things we learn in our lives let us do.
But then there's this whole layer of kind of formalization
that got developed in which is the kind of whole
sort of story of intellectual history
and whole kind of depth of learning
that formalization turned into things like logic,
mathematics, science and so on.

And that's the kind of thing that allows one
to kind of build these towers of things you work out.
It's not just I can immediately figure this out.
It's no, I can use this kind of formalism
to go step by step and work out something
which was not immediately obvious to me.
And that's kind of the story of what we're trying
to do computationally is to be able to build
those kind of tall towers of what implies,
what implies, what and so on.
And as opposed to kind of the,
yes, I can immediately figure it out.
It's just like what I saw somewhere else
in something that I heard or remembered
or something like this.
What can you say about the kind of formal structure,
the kind of formal foundation you can build
such a formal structure on?
About the kinds of things you would start on
in order to build this kind of deep
computable knowledge trees.
So the question is sort of how do you think
about computation?
And there's a couple of points here.
One is what computation intrinsically is like
and the other is what aspects of computation
we humans with our minds and with the kinds
of things we've learnt can sort of relate to
in that computational universe.
So if we start on the kind of what can computation
we like, it's something I've spent
some big chunk of my life studying,
is imagine that you're, you know,
we usually write programs where we kind of know
what we want the program to do
and we carefully write many lines of code
and we hope that the program does what we intended it to do.
But the thing I've been interested in is
if you just look at the kind of natural science of programs,
you just say, I'm gonna make this program,
it's a really tiny program.
Maybe I even pick the pieces of the program at random,
but it's really tiny.

I really tiny, I mean, you know,
less than a line of code type thing.
You say, what does this program do?
And you run it and big discovery that I made
in the early 80s is that even extremely simple programs
when you run them can do really complicated things.
Really surprised me, it took me several years
to kind of realize that that was a thing, so to speak.
But that realization that even very simple programs
can do incredibly complicated things
that we very much don't expect.
That discovery, I mean, I realized that that's very much,
I think, how nature works.
That is nature has simple rules,
but yet does all sorts of complicated things
that we might not expect.
You know, as big thing of the last few years
has been understanding that that's how the whole universe
and physics works, but that's a quite separate topic.
But so there's this whole world of programs
and what they do and very rich, sophisticated things
that these programs can do.
But when we look at many of these programs,
if you look at them, you say, well, that's kind of,
I don't really know what that's doing.
It's not a very human kind of thing.
So on the one hand, we have sort of what's possible
in the computational universe.
On the other hand, we have the kinds of things
that we humans think about,
the kinds of things that are developed
in kind of our intellectual history.
And that's really the challenge
to sort of making things computational
is to connect what's computationally possible
out in the computational universe
with the things that we humans
sort of typically think about with our minds.
Now, that's a complicated kind of moving target
because the things that we think about change over time.
We've learned more stuff.
We've invented mathematics.
We've invented various kinds of ideas

and structures and so on.
So it's gradually expanding.
We're kind of gradually colonizing more and more
of this kind of intellectual space of possibilities.
But the real thing, the real challenge is,
how do you take what is computationally possible?
How do you encapsulate the kinds of things
that we think about in a way that kind of plugs in
to what's computationally possible?
And actually, the big sort of idea there
is this idea of kind of symbolic programming,
symbolic representations of things.
And so the question is,
when you look at sort of everything in the world
and you kind of take some visual scene
or something you're looking at,
and you say, well, how do I turn that into something
that I can kind of stuff into my mind?
There are lots of pixels in my visual scene,
but the things that I remembered from that visual scene
are there's a chair in this place.
It's a kind of a symbolic representation
of the visual scene.
There are two chairs and a table or something,
rather than there are all these pixels arranged
in all these detailed ways.
And so the question then is,
how do you take sort of all the things in the world
and make some kind of representation
that corresponds to the types of ways
that we think about things?
And human language is sort of one form of representation
that we have.
We talk about chairs, that's a word in human language
and so on.
How do we take, but human language is not
in and of itself something that plugs in very well
to sort of computation.
It's not something from which
you can immediately compute consequences and so on.
And so you have to kind of find a way
to take sort of the stuff we understand
from human language and make it more precise.

And that's really the story of symbolic programming
and what that turns into is something
which I didn't know at the time
it was going to work as well as it has.
But back in the 1979 or so,
I was trying to build my first big computer system
and trying to figure out,
how should I represent computations at a high level?
And I kind of invented this idea of using
kind of symbolic expressions structured
as it's kind of like a function and a bunch of arguments.
But that function doesn't necessarily evaluate to anything.
It's just a thing that sits there representing a structure.
And so building up that structure
and it's turned out that structure has been extremely,
it's a good match for the way that we humans,
it seems to be a good match for the way that we humans
kind of conceptualize higher level things.
And it's been for the last, I don't know,
45 years or something.
It's served me remarkably well.
So building up that structure
using this kind of symbolic representation.
But what can you say about abstractions here?
Because you could just start with your physics project,
you could start at a hypergraph at a very, very low level
and build up everything from there, but you don't.
You take shortcuts.
Right.
You take the highest level of abstraction,
convert that, the kind of abstraction
that's convertible to something computable
using symbolic representation.
And then that's your new foundation
for that little piece of knowledge.
And somehow all of that is integrated.
Right.
So the sort of a very important phenomenon
that is kind of a thing that I've sort of realized
is just, it's one of these things that sort of
in the future of kind of everything
is going to become more and more important
is this phenomenon of computational irreducibility.

And the question is, if you know the rules for something,
you have a program, you're going to run it,
you might say, I know the rules.
Great, I know everything about what's going to happen.
Well, in principle, you do,
because you can just run those rules out
and just see what they do.
You might run them a million steps,
you see what happens, et cetera.
The question is, can you like immediately jump ahead
and say, I know what's going to happen
after a million steps and the answer is 13 or something.
And one of the very critical things to realize is
if you could reduce that computation,
there isn't a sense, no point in doing the computation.
The place where you really get value out of doing computation
is when you had to do the computation
to find out the answer.
But this phenomenon that you have to do the computation
to find out the answer,
this phenomenon of computational irreducibility
seems to be tremendously important
for thinking about lots of kinds of things.
So one of the things that happens is,
okay, you've got a model of the universe at the low level
in terms of atoms of space and hypergraphs
and rewriting of hypergraphs and so on.
And it's happening 10 to the 100 times every second,
let's say, well, you say, great, then we've nailed it.
We know how the universe works.
Well, the problem is the universe
can figure out what it's going to do.
It does those 10 to the 100 steps,
but for us to work out what it's going to do,
we have no way to reduce that computation.
The only way to do the computation,
to see the result of the computation is to do it.
And if we're operating within the universe,
there's no opportunity to do that
because the universe is doing it
as fast as the universe can do it
and that's what's happening.
So what we're trying to do,

and a lot of the story of science
and a lot of other kinds of things
is finding pockets of reducibility.
That is, you could have a situation
where everything in the world
is full of computational irreducibility.
We never know what's going to happen next.
The only way we can figure out what's going to happen next
is just let the system run and see what happens.
So in a sense, the story of most kinds of science,
inventions, a lot of kinds of things
is the story of finding these places
where we can locally jump ahead.
And one of the features of computational irreducibility
is there are always pockets of reducibility.
There are always places,
there are always an infinite number of places
where you can jump ahead.
There's no way where you can jump completely ahead,
but there are little patches,
little places where you can jump ahead a bit.
And I think we can talk about physics project and so on,
but I think the thing we realize is
we kind of exist in a slice
of all the possible computational irreducibility
in the universe.
We exist in a slice
where there's a reasonable amount of predictability.
And in a sense, as we try and construct
these kind of higher levels of abstraction,
symbolic representations and so on,
what we're doing is we're finding these lumps
of reducibility that we can kind of attach ourselves to,
and about which we can kind of have
fairly simple narrative things to say.
Because in principle, I say,
what's gonna happen in the next few seconds?
Oh, there are these molecules moving around
in the air in this room,
and oh gosh, it's an incredibly complicated story.
And that's a whole computational irreducible thing,
most of which I don't care about.
And most of it is, well, the air's still gonna be here

and nothing much is going to be different about it.
And that's a kind of reducible fact
about what is ultimately at an underlying level
of the computational irreducible process.
And life would not be possible
if we didn't have a large number of such reducible pockets.
Yes.
Pockets amenable to reduction into something symbolic.
Yes, I think so.
I mean, life in the way that we experience it,
that, I mean, one might,
depending on what we mean by life, so to speak,
the experience that we have
of sort of consistent things happening in the world,
the idea of space, for example,
where there's, you know, we can just say,
you're here, you move there.
It's kind of the same thing.
It's still you in that different place,
even though you're made of different atoms of space and so on.
This is this idea that it's,
that there's sort of this level of predictability
of what's going on.
That's us finding a slice of reducibility
in what is underneath this computationally irreducible
kind of system.
And I think that's sort of the thing
which is actually my favorite discovery
over the last few years is the realization
that it is sort of the interaction
between the sort of underlying computational irreducibility
and our nature as kind of observers
who sort of have to key into computational reducibility.
That fact leads to the main laws of physics
that we discovered in the 20th century.
So this is, we talk about this in more detail,
but this is, to me, it's kind of our nature as observers,
the fact that we are computationally bounded observers,
we don't get to follow all those little pieces
of computational irreducibility
to stuff what is out there in the world into our minds
requires that we are looking at things
that are reducible, we are compressing,

kind of we're extracting just some essence,
some kind of symbolic essence of what's the detail
of what's going on in the world,
that together with one other condition
that at first seems sort of trivial,
but isn't, which is that we believe
we are persistent in time.
That is, you know.
So sometimes the causality.
Here's the thing, at every moment, according to our theory,
we're made of different atoms of space.
At every moment, sort of the microscopic detail
of what the universe is made of
is being rewritten.
And that's, and in fact, the very fact
that there's coherence between different parts of space
is a consequence of the fact
that there are all these little processes going on
that kind of knit together the structure of space.
It's kind of like, if you wanted to have a fluid
with a bunch of molecules in it,
if those molecules weren't interacting,
you wouldn't have this fluid that would pour
and do all these kinds of things.
It would just be sort of a free-floating
collection of molecules.
So similarly it is with space,
that the fact that space is kind of knitted together
is a consequence of all this activity in space.
And the fact that kind of what we consist of
sort of this series of, we're continually being rewritten.
And the question is, why is it the case
that we think of ourselves as being the same us through time?
That's kind of a key assumption.
I think it's a key aspect of what we see
as sort of our consciousness, so to speak,
is that we have this kind of consistent threat of experience.
Well, isn't that just another limitation of our mind
that we want to reduce reality into some,
that kind of temporal consistency
is just a nice narrative to tell ourselves?
Well, the fact is, I think it's critical
to the way we humans typically operate

is that we have this single thread of experience.
If you imagine sort of a mind where you have,
maybe that's what's happening in various kinds of minds
that aren't working the same way other minds work,
is that you're splitting into multiple threads of experience.
It's also something where, when you look at,
I don't know, quantum mechanics, for example,
in the insides of quantum mechanics,
it's splitting into many threads of experience.
But in order for us humans to interact with it,
you kind of have to knit all those different threads together
so that we say, oh yeah, a definite thing happened,
and now the next definite thing happens and so on.
And I think, sort of inside,
it's sort of interesting to try and imagine
what's it like to have kind of these
fundamentally multiple threads of experience going on?
I mean, right now, different human minds
have different threads of experience.
We just have a bunch of minds
that are interacting with each other,
but we don't have a, within each mind,
there's a single thread,
and that is indeed a simplification.
I think it's a thing,
the general computational system
does not have that simplification.
And it's one of the things,
people often seem to think that consciousness
is the highest level of kind of things
that can happen in the universe, so to speak.
But I think that's not true.
I think it's actually a specialization
in which, among other things,
you have this idea of a single thread of experience,
which is not a general feature of anything
that could kind of computationally happen in the universe.
So it's a feature of a computationally limited system
that's only able to observe reducible pockets.
So, I mean, this word observer,
it means something in quantum mechanics.
It means something in a lot of places.
It means something to us humans as conscious beings.

So what's the importance of the observer?
What is the observer?
What's the importance of the observer
in the computational universe?
So this question of what is an observer?
What's the general idea of an observer?
It's actually one of my next projects,
which got somewhat derailed by the current sort of AI mania,
but-
Is there a connection there or is that,
do you think the observer is primarily a physics phenomena?
Is it related to the whole AI thing?
Yes. Yes, it is related.
So one question is, what is a general observer?
So, you know, we know, we have an idea
what is a general computational system.
We think about Turing machines.
We think about other models of computation.
There's a question, what is a general model of an observer?
And there's kind of observers like us,
which is kind of the observers we're interested in.
You know, we could imagine an alien observer
that deals with computational irreducibility
and it has a mind that's utterly different from ours
and completely incoherent with what we're like.
But the fact is that, you know,
if we are talking about observers like us,
that one of the key things is this idea
of kind of taking all the detail of the world
and being able to stuff it into a mind,
being able to take all the detail
and kind of, you know, extract out of it
a smaller set of kind of degrees of freedom,
a smaller number of elements
that will sort of fit in our minds.
And I think this question, so I've been interested
in trying to characterize what is the general observer?
And the general observer is, I think,
in part there are many, let me give an example of a,
you know, you have a gas,
it's got a bunch of molecules bouncing around.
And the thing you're measuring about the gas
is its pressure.

And the only thing you as an observer care about is pressure.
And that means you have a piston on the side of this box
and the piston is being pushed by the gas.
And there are many, many different ways
that molecules can hit that piston.
But all that matters is the kind of aggregate
of all those molecular impacts
because that's what determines pressure.
So there's a huge number of different configurations
of the gas, which are all equivalent.
So I think one key aspect of observers
is this equivalencing of many different configurations
of a system saying all I care about is this aggregate feature.
All I care about is this overall thing.
And that's sort of one aspect.
And when we see that in lots of different,
again, it's the same story over and over again,
that there's a lot of detail in the world,
but what we are extracting from it
is something sort of a thin summary of that detail.
Is that thin summary, nevertheless true?
Can it be a crappy approximation?
That on average is correct.
I mean, if we look at the observer that's the human mind,
it seems like there's a lot of very,
as represented by natural language, for example,
there's a lot of really crappy approximation.
Sure.
And that could be maybe a feature of it.
Well, yes.
There's ambiguity.
Right, right.
You don't know, it could be the case.
You're just measuring the aggregate impact
to these molecules,
but there is some tiny, tiny probability
that the molecules will arrange themselves
in some really funky way.
And that just measuring that average
isn't going to be the main point.
By the way, an awful lot of science
is very confused about this,
because you look at papers and people are really keen,

they draw this curve and they have these bars
on the curve and things, and it's just this curve.
And it's this one thing,
and it's supposed to represent some system
that has all kinds of details in it.
And this is a way that lots of science has gotten wrong,
because people say, I remember years ago,
I was studying snowflake growth.
You have a snowflake and it's growing,
it has all these arms, it's doing complicated things,
but there was a literature on this stuff
and it talked about what's the rate of snowflake growth?
And it got pretty good answers
for the rate of the growth of the snowflake.
And they looked at it more carefully,
and they had these nice curves
of snowflake growth rates and so on.
I looked at it more carefully and I realized,
according to their models, the snowflake will be spherical.
And so they got the growth rate right,
but the detail was just utterly wrong.
And not only the detail,
the whole thing was not capturing,
it was capturing this aspect of the system
that was in a sense missing the main point
of what was going on.
What is the geometric shape of a snowflake?
Snowflakes start in the phase of water
that's relevant to the formation of snowflakes.
It's a phase of ice,
which starts with a hexagonal arrangement of water molecules.
And so it starts off growing as a hexagonal plate.
And then what happens is...
It's a plate, oh, versus sphere.
Well, no, no, but it's much more than that.
I mean, snowflakes are fluffy.
Typical snowflakes have little dendritic arms.
And what actually happens is it's kind of cool
because you can make these very simple discrete models
with cellular automata and things that figure this out.
You start off with this hexagonal thing,
and then the places, it starts to grow little arms.
And every time a little piece of ice

that adds itself to the snowflake,
the fact that that ice condensed from the water vapor
heats the snowflake up locally.
And so it makes it less likely for another piece of ice
to accumulate right nearby.
So this leads to a kind of growth inhibition.
So you grow an arm, and it is a separated arm
because right around the arm, it got a little bit hot
and it didn't add more ice there.
So what happens is it grows, you have a hexagon,
it grows out arms, the arms grow arms,
and then the arms grow, arms grow arms,
and eventually, actually it's kind of cool
because it actually fills in another hexagon,
a bigger hexagon.
And when I first looked at this,
you had a very simple model for this,
I realized when it fills in that hexagon,
it actually leaves some holes behind.
So I thought, well, is that really right?
So I look at these pictures of snowflakes
and sure enough, they have these little holes in them
that are kind of scars of the way that these arms grow out.
So you can't fill in backfill holes.
So you just keep going up.
They don't backfill, yeah, they don't backfill.
And presumably there's a limitation on how big,
like you can't arbitrarily grow.
I'm not sure, I mean, the thing falls through the,
I mean, I think it does, you know,
it hits the ground at some point.
I think you can grow, I think you can grow in the lab,
I think you can grow pretty big ones.
I think you can grow many iterations
of this kind of goes from hexagon, it grows out arms,
it turns back, it fills back into a hexagon,
it grows more arms again.
In 3D.
No, it's flat usually.
Boy, why is it flat?
Why doesn't it span out?
Okay, okay, wait a minute.
You said it's fluffy and fluffy

is a three-dimensional property, no?
No, it's fluffy.
Snow is, okay, so, you know, what makes,
we're really, we're really in it.
I like this, let's go there.
It's multiple snowflakes become fluffy.
What single snowflake is not fluffy?
No, no, a single snowflake is fluffy.
And what happens is, you know, if you have snow
that is just pure hexagons,
they can, you know, they fit together pretty well.
It's not, it doesn't make,
it doesn't have a lot of air in it.
And they can also slide against each other pretty easily.
And so the snow can be pretty, you know,
can, I think avalanches happen sometimes
when the things tend to be these, you know,
hexagonal plates and it kind of slides.
But then when the thing has all these arms
that have grown out, it's not,
they don't fit together very well.
And that's why the snow has lots of air in it.
And if you look at one of these snowflakes,
if you catch one, you'll see it has these little arms.
And people, actually people often say, you know,
no two snowflakes are alike.
That's mostly because as a snowflake grows,
they do grow pretty consistently
with these different arms and so on.
But you capture them at different times.
As they, you know, they fell through the air
in a different way.
You'll catch this one at this stage.
And as it goes through different stages,
they look really different.
And so that's why, you know,
it kind of looks like no two snowflakes are alike
because you caught them at different times.
So the rules under which they grow are the same.
It's just the timing is, okay.
So the point is science is not able to describe
the full complexity of snowflake growth.
Well, science, if you do what people might often do,

which is say, okay, let's make it scientific.
Let's turn into one number.
And that one number is kind of the growth rate of the arms
or something such other thing.
That fails to capture sort of the detail
of what's going on inside the system.
And that's in a sense a big challenge for science
is how do you extract from the natural world?
For example, those aspects of it
that you are interested in talking about.
Now you might just say,
I don't really care about the fluffiness of the snowflakes.
All I care about is the growth rate of the arms.
In which case, you know, you have,
you can have a good model
without knowing anything about the fluffiness.
But the fact is as a practical, you know,
if you say what's the,
what is the most obvious feature of a snowflake?
Oh, that it has this complicated shape.
Well, then you've got a different story about what you model.
I mean, this is one of the features
of sort of modeling in science, that you know,
what is a model?
A model is some way of reducing the actuality of the world
to something where you can readily sort of give a narrative
for what's happening,
where you can basically make some kind of abstraction
of what's happening and answer questions
that you care about answering.
If you wanted to answer all possible questions
about the system, you'd have to have the whole system
because you might care about this particular molecule.
Where did it go?
And, you know, your model,
which is some big abstraction of that,
has nothing to say about that.
So, you know, one of the things
that's often confusing in science is people will have,
I've got a model, somebody says.
Somebody else will say, I don't believe in your model
because it doesn't capture the feature of the system
that I care about.

You know, there's always this controversy about, you know,
is it a correct model?
Well, no model is a, except for the actual system itself,
is a correct model in the sense that it captures everything.
Questions doesn't capture what you care about capturing.
Sometimes that's ultimately defined
by what you're going to build technology out of,
things like this.
The one counter example to this is,
if you think you're modeling the whole universe
all the way down, then there is a notion of a correct model.
But even that is more complicated
because it depends on kind of how observers sample things
and so on, that's a separate story.
But at least at the first level, to say, you know,
this thing about, oh, it's an approximation,
you're capturing one aspect,
you're not capturing other aspects.
When you really think you have a complete model
for the whole universe,
you better be capturing ultimately everything,
even though to actually run that model is impossible
because of computational irreducibility.
The only thing that successfully runs that model
is the actual running of the universe.
Is the universe itself.
But okay, so what you care about is an interesting concept.
So that's a human concept.
So that's what you're doing with Wolfram Alpha
and Wolfram Language,
is you're trying to come up with symbolic representations.
Yes.
As simple as possible.
So a model that's as simple as possible
that fully captures stuff we care about.
Yes.
So I mean, for example, you know,
we could, we'll have a thing about, you know,
data about movies, let's say.
We could be describing every individual pixel
in every movie and so on,
but that's not the level that people care about.
And it's, yes, this is a, I mean,

and that level that people care about
is somewhat related to what's described in natural language.
But what we're trying to do is to find a way
to sort of represent precisely so you can compute things.
See, one thing when you say,
you give a piece of natural language,
question is you feed it to a computer.
You say, does the computer understand this natural language?
Well, you know, the computer processes it in some way,
it does this, maybe it can make a continuation
of the natural language, you know,
maybe it can go on from the prompt
and say what it's gonna say.
You say, does it really understand it?
Hard to know.
But for in this kind of computational world,
there is a very definite definition of does it understand?
Which is, could it be turned into this symbolic
computational thing from which you can compute
all kinds of consequences?
And that's the sense in which one has sort of a target
for the understanding of natural language.
And that's kind of our goal is to have as much as possible
about the world that can be computed in a reasonable way,
so to speak, be able to be sort of captured
by this kind of computational language.
That's kind of the goal.
And I think for us humans,
the main thing that's important is,
as we formalize what we're talking about,
it gives us a way of kind of building a structure
where we can sort of build this tower
of consequences of things.
So if we're just saying,
well, let's talk about it in natural language,
it doesn't really give us some hard foundation
that lets us build step by step to work something out.
I mean, it's kind of like what happens in math.
If we were just sort of vaguely talking about math,
but didn't have the kind of full structure of math
and all that kind of thing,
we wouldn't be able to build this kind of big tower
of consequences.

And so, in a sense, what we're trying to do
with the whole computational language effort
is to make a formalism for describing the world
that makes it possible to kind of build
this tower of consequences.
Well, can you talk about this dance
between natural language and Wolfram language?
So there's this gigantic thing called the internet
where people post memes and diary type thoughts
and very important sounding articles and all of that
that makes up the training data set for GPT.
And then there's a Wolfram language.
How can you map from the natural language of the internet
to the Wolfram language?
Is there an manual?
Is there an automated way of doing that
as we look into the future?
Well, so Wolfram Alpha, what it does,
it's kind of front end is turning natural language
into computational language.
What you mean by that is there's a prompt,
you ask a question, what is the capital of some...
And it turns into what's the distance
between Chicago and London or something.
And that will turn into geodistance of entity, city,
et cetera, et cetera, et cetera.
Each one of those things is very well-defined.
We know, given that it's the entity, city, Chicago,
et cetera, et cetera, et cetera, Illinois, United States,
we know the geolocation of that,
we know its population,
we know all kinds of things about it,
which we have curated that data
to be able to know that with some degree of certainty,
so to speak.
And then we can compute things from this.
And that's kind of the, yeah, that's the idea.
But then something like GPT, large language models,
do they allow you to make that conversion much more powerful?
Okay, so that's an interesting thing,
which we still don't know everything about, okay?
The, I mean, this question of going from natural language
to computational language.

Yes.
In Wolfram Alpha, we've now,
Wolfram Alpha's been out and about
for what, 13 and a half years now.
And we've achieved, I don't know what it is,
98%, 99% success on queries that get put into it.
Now, obviously there's a sort of feedback loop
because the things that work
are things people go on putting into it.
So that, but we've got to a very high success rate
of the little fragments of natural language
that people put in.
Questions, math calculations, chemistry calculations,
whatever it is, we can, we do very well at that,
turning those things into computational language.
Now, from the very beginning of Wolfram Alpha,
I thought about, for example,
writing code with natural language.
In fact, I had, I was just looking at this recently,
I had a post that I wrote in 2010, 2011,
called something like programming with natural language
is actually going to work, okay?
And so, we had done a bunch of experiments
using methods that were a little bit,
some of them a little bit machine learning like,
but certainly not the same kind of idea
of vast training data and so on,
that is the story of large language models.
Actually, I know that that post,
a piece of utter trivia, but that post,
Steve Jobs forwarded that post around
to all kinds of people at Apple.
And he didn't know that was,
because he never really liked programming languages.
So he was very happy to see the idea
that you could get rid of this kind of layer
of kind of engineering-like structure.
He would have liked, I think, what's happening now,
because it really is the case that you can,
this idea that you have to kind of learn
how the computer works to use a programming language
is something that is, I think, a thing that,
just like you had to learn the details of the op codes

to know how assembly language worked and so on,
it's kind of a thing that's a limited time horizon.
But kind of the, so this idea of how elaborate
can you make kind of the prompt,
how elaborate can you make the natural language
and abstract from it computational language?
It's a very interesting question.
And what chat GBT4 and so on can do is pretty good.
It's very interesting process.
I mean, I'm still trying to understand this workflow.
We've been working out a lot of tooling
around this workflow that's pretty interesting.
The natural language to computational language.
The process, especially if it's conversation,
like dialogue, it's like multiple queries kind of thing.
Yeah, right.
There's so many things that are really interesting,
that work and so on.
So first thing is, can you just walk up to the computer
and expect to sort of specify computation?
What one realizes is humans have to have some idea
of kind of this way of thinking about things computationally.
Without that, you're kind of out of luck,
because you just have no idea
what you're going to walk up to a computer.
I remember when I should tell a silly story about myself,
the very first computer I saw,
which was when I was 10 years old,
and it was a big mainframe computer and so on,
and I didn't really understand what computers did.
And it's like somebody was showing me this computer
and it's like, you know,
can the computer work out the weight of a dinosaur?
It's like, that isn't a sensible thing to ask.
That's kind of, you know, you have to give it,
that's not what computers do.
I mean, in Wolfman, for example,
you could say, what's the typical weight of a stegosaurus?
It will give you some answer,
but that's a very different kind of thing
from what one thinks of computers as doing.
And so the kind of the question of, you know,
first thing is people have to have an idea

of what computation is about.
You know, I think it's a very, you know, for education,
that is the key thing.
It's kind of this notion, not computer science,
not so the details are programming,
but just this idea of how do you think
about the world computationally?
Computation, thinking about the world computationally
is kind of this formal way of thinking about the world.
We've had other ones, like logic was a formal way,
you know, as a way of sort of abstracting
and formalizing some aspects of the world.
Mathematics is another one.
Computation is this very broad way
of sort of formalizing the way we think about the world.
And the thing that's cool about computation is
if we can successfully formalize things
in terms of computation,
computers can help us figure out what the consequences are.
It's not like you formalized it with math,
well, that's nice, but now you have to,
if you're not using a computer to do the math,
you have to go work out a bunch of stuff yourself.
So I think, but this idea, let's see,
I mean, we're trying to take kind of the,
we're talking about sort of natural language
and its relationship to computational language.
The thing, sort of the typical workflow, I think,
is first, human has to have some kind of idea
of what they're trying to do.
That if it's something that they want to sort of build
a tower of capabilities on,
something that they want to sort of formalize
and make computational.
So then human can type something in to, you know,
some LLM system and sort of say vaguely what they want.
In sort of computational terms,
then it does pretty well at synthesizing
Wolfram Language Code.
And it'll probably do better in the future
because we've got a huge number of examples
of natural language input together
with the Wolfram Language translation of that.

So it's kind of a, you know,
that's a thing where you can kind of extrapolating
from all those examples makes it easier to do that task.
Is the prompter task could also kind of debugging
the Wolfram Language Code?
Or is your hope to not do that debugging?
Oh, no, no, no.
I mean, so there are many steps here.
So the first thing is you type natural language.
It generates Wolfram Language Code.
You have examples, by the way.
Do you have an example that, is it the dinosaur example?
Do you have an example that jumps to mind
that we should be thinking about some dumb example?
It's like, take my heart rate data
and, you know, figure out whether I, you know,
make a moving average every seven days or something
and work out what the, and make a plot of the result.
Okay?
So that's a thing which is, you know,
about two-thirds of a line of Wolfram Language Code.
I mean, it's, you know,
list plot of moving average of some data bin
or something of the data,
and then you'll get the result.
And, you know, the vague thing
that I was just saying in natural language
could, would, almost certainly,
turn into that very simple piece of Wolfram Language Code.
So you start mumbling about heart rate.
Yeah.
And kind of, you know, you arrive
at the moving average kind of idea.
Right.
You say average over seven days.
Maybe it'll figure out that that's a moving, you know,
that that can be encapsulated as this moving average idea.
I'm not sure.
But then the typical workflow that I'm seeing is,
you generate this piece of Wolfram Language Code.
It's pretty small, usually.
It's, and if it isn't small, it probably isn't right.
But, you know, if it's, it's pretty small and, you know,

Wolfram Language is one of the ideas of Wolfram Language
is it's a language that humans can read.
It's not a language which, you know,
programming languages tend to be this one-way story
of humans write them and computers execute from them.
Wolfram Language is intended to be something
which is sort of like math notation,
something where, you know, humans write it
and humans are supposed to read it as well.
And so kind of the workflow that's emerging
is kind of this, this human mumble some things,
you know, large language model produces
a fragment of Wolfram Language Code.
Then you look at that, you say, you know, that looks,
well, typically you just run it first.
You see, does it produce the right thing?
You look at what it produces.
You might say that's obviously crazy.
You look at the code.
You see, I see why it's crazy.
You fix it.
If you really care about the result
and you really want to make sure it's right,
you need to look at that code and understand it
because that's the way you have the sort of checkpoint
of did it really do what I expected it to do.
Now, you go beyond that.
I mean, it's, you know, what we find is, for example,
let's say the code does the wrong thing.
Then you can often say to the large language model,
can you adjust this to do this?
And it's pretty good at doing that.
Interesting.
So you're using the output of the code
to give you hints about the function of the code.
So you're debugging based on the output of the code,
not the code itself.
Right.
The plugin that we have, you know, for chat GPT,
it does that routinely.
You know, it will send the thing in,
it will get a result, it will discover,
the LLM will discover itself that the result is not plausible

and it will go back and say, oh, I'm sorry,
it's very polite and it, you know,
it goes back and says I'll rewrite that piece of code
and then it will try it again and get the result.
The other thing that's pretty interesting is
when you're just running, so one of the new concepts
that we have, we invented this whole idea of notebooks
back 36 years ago now.
And so now there's the question of sort of
how do you combine this idea of notebooks where you have,
you know, text and code and output.
How do you combine that with the notion of chat and so on?
And there's some really interesting things there.
Like for example, a very typical thing now
is we have these notebooks where as soon as the,
if the thing produces errors, if they, you know,
run this code and it produces messages and so on,
the LLM automatically not only looks at those messages,
it can also see all kinds of internal information
about stack traces and things like this.
And it can then, it does a remarkably good job
of guessing what's wrong and telling you.
So in other words, it's looking at things,
it's sort of interesting.
It's kind of a typical sort of AI-ish thing
that it's able to have more sensory data
than we humans are able to have,
because they're able to look at a bunch of stuff
that we humans would kind of glaze over looking at
and it's able to then come up with,
oh, this is the explanation of what's happening.
And what is the data, the stack trace,
the code you've written previously,
the natural language you've written?
Yeah, it's also what's happening is one of the things
that is, for example, when there's these messages,
there's documentation about these messages,
there's examples of where the messages have occurred,
otherwise, all these kinds of things.
The other thing that's really amusing with this
is when it makes a mistake,
one of the things that's in our prompt
when the code doesn't work is, read the documentation.

And we have another piece of the plugin
that lets it read documentation.
And that, again, is very, very useful,
because it will figure out sometimes it'll get,
it'll make up the name of some option for some function
that doesn't really exist, read the documentation,
it'll have some wrong structure for the function and so on.
That's a powerful thing.
I mean, the thing that I've realized is
we built this language over the course of all these years
to be nice and coherent and consistent and so on,
so it's easy for humans to understand.
Turns out there was a side effect that I didn't anticipate,
which is it makes it easy for AIs to understand.
So it's almost like another natural language.
Yeah.
So Wolfram language is a kind of foreign language.
Yes.
You have a lineup.
English, French, Japanese, Wolfram language,
and then, I don't know, Spanish,
and then the system is not going to notice, hopefully.
Well, yes.
I mean, maybe.
You know, that's an interesting question,
because it really depends on what I see as being
an important piece of fundamental science
that basically just jumped out at us with ChatGPT.
Because I think, you know, the real question is,
why does ChatGPT work?
How is it possible to encapsulate, you know,
to successfully reproduce all these kinds of things
in natural language, you know, with a, you know,
a comparatively small, he says, you know,
a couple of hundred billion, you know,
weights of neural net and so on.
And I think that, you know, that relates to kind of
a fundamental fact about language, which, you know,
that the main thing is that I think there's a structure
that we haven't kind of really explored very well.
It's kind of this semantic grammar I'm talking about,
about language.
I mean, we kind of know that when we set up human language,

we know that it has certain regularities.
We know that it has a certain grammatical structure,
you know, noun followed by verb, followed by noun,
adjectives, et cetera, et cetera, et cetera.
That's its kind of grammatical structure.
But I think the thing that ChatGPT is showing us
is that there's an additional kind of regularity to language
which has to do with the meaning of the language
beyond just this pure, you know,
part of speech combination type of thing.
And I think the kind of the one example of that
that we've had in the past is logic.
And, you know, I think my sort of kind of picture
of how was logic invented?
How was logic discovered?
It really was the thing that was discovered
in its original conception.
It was discovered, presumably by Aristotle,
who kind of listened to a bunch of people, orators,
you know, giving speeches.
And this one made sense, that one doesn't make sense.
This one, and, you know, you see these patterns of, you know,
if the, you know, I don't know what, you know,
if the Persians do this, then the this does that,
et cetera, et cetera, et cetera.
And what Aristotle realized is there's a structure
to those sentences, there's a structure to that rhetoric
that doesn't matter whether it's the Persians and the Greeks
or whether it's the cats and the dogs.
It's just, you know, P and Q.
You can abstract from the details
of these particular sentences.
You can lift out this kind of formal structure,
and that's what logic is.
That's a heck of a discovery, by the way, logic.
You're making me realize, no.
Yeah.
It's not obvious.
The fact that there is an abstraction from natural language
that has where you can fill in any word you want
is a very interesting discovery.
Now, it took a long time to mature.
I mean, Aristotle had this idea of syllogistic logic

where there were these particular patterns
of how you could argue things, so to speak.
And, you know, in the Middle Ages,
part of education was you memorize the syllogisms.
I forget how many there were, but 15 of them or something.
And they all had names.
They all had mnemonics, like I think Barbara and Sellerant
were two of the mnemonics for the syllogisms.
And people would kind of, this is a valid argument
because it follows the Barbara syllogism, so to speak.
And it took until 1830, you know, with George Boole
to kind of get beyond that and kind of see that
there was a level of abstraction that was beyond
this particular template of a sentence, so to speak.
And what's interesting there is, in a sense,
chatGBT is operating at the Aristotelian level.
It's essentially dealing with templates of sentences.
By the time you get to Boole and Boolean Algebra
and this idea of, you know, you can have arbitrary depth nested
collections of ands and auras and knots
and you can resolve what they mean,
that's the kind of thing, that's a computation story.
You've gone beyond the pure sort of templates of natural language
to something which is an arbitrarily deep computation.
But the thing that I think we realize from chatGBT
is, you know, Aristotle stopped too quickly
and there was more that you could have lifted
out of language as formal structures.
And I think there's, you know, in a sense,
we've captured some of that and, you know,
some of what is in language, there's a lot of kind of
little calculi, little algebras of what you can say,
what language talks about.
I mean, whether it's, I don't know, if you say,
I go from place A to place B, place B to place C,
then I know I've gone from place A to place C.
If A is a friend of B and B is a friend of C,
it doesn't necessarily follow that A is a friend of C.
These are things that are, you know,
if you go from place A to place B, place B to place C,
it doesn't matter how you went, like logic,
it doesn't matter whether you flew there,
walked there, swam there, whatever.

You still, this transitivity of where you go is still valid.
And there are many kinds of kind of features, I think,
of the way the world works that are captured
in these aspects of language, so to speak.
And I think what ChatGBT effectively has found,
just like it discovered logic, you know,
people are really surprised it can do these logical inferences.
It discovered logic the same way Aristotle discovered logic
by looking at a lot of sentences effectively
and noticing the patterns in those sentences.
But it feels like it's discovering something
much more complicated than logic.
So this kind of semantic grammar,
I think you wrote about this,
maybe you can call it the laws of language,
I believe you call, or which I like, the laws of thought.
Yes.
That was the title that George Boole had
for his Boolean algebra back in 1830.
Laws of thought.
Yes, that was what he said.
All right.
So he thought he nailed it with Boolean algebra.
Yeah.
There's more to it.
And it's a good question of how much more is there to it.
And it seems like one of the reasons as you imply
that the reason GPT works, ChatGBT works,
is that there's a finite number of things to it.
Yeah, I mean, it's discovering the laws.
In some sense, GPT is discovering the laws of semantic grammar
that underlies language.
Yes.
And what's sort of interesting is in the computational universe,
there's a lot of other kinds of computation that you could do.
They're just not ones that we humans have cared about
and operate with.
And that's probably because our brains are built in a certain way.
And the neural nets of our brains are not that different,
in some sense, from the neural nets of a large language model.
And that's kind of, and so when we think about,
and maybe we can talk about this some more,
but when we think about sort of what will AIs ultimately do?

The answer is, insofar as AIs are just doing computation,
they can run off and do all these kinds of crazy computations.
But the ones that we sort of have decided we care about
is this kind of very limited set.
That's where the reinforcement learning with human feedback
seems to come in.
The more the AIs say the stuff that kind of interests us,
the more we're impressed by it.
So it can do a lot of interesting intelligent things,
but we're only interested in the AI systems
when they communicate in a human-like way.
Yes.
About human-like topics.
Yes.
Well, it's like technology.
I mean, in a sense, the physical world provides all kinds of things.
There are all kinds of processes going on in physics.
Only a limited set of those are ones that we capture
and use for technology, because they're only a limited set
where we say, this is a thing that we can sort of apply
to the human purposes we currently care about.
I mean, you might have said,
okay, you pick up a piece of rock.
You say, okay, this is a nice silicate.
It contains all kinds of silicon.
I don't care.
Then you realize, oh, we could actually turn this into
a semiconductor wafer and make it microprocessor out of it,
and then we care a lot about it.
Yes.
And it's this thing about what do we...
In the evolution of our civilization,
what things do we identify as being things we care about?
I mean, it's like when there was a little announcement
recently of the possibility of a high-temperature superconductor
that involved the element lutetium,
which generally nobody has cared about.
It's kind of...
But suddenly, if there's this application
that relates to kind of human purposes,
we start to care a lot.
So, given your thinking that GPT may have discovered
in clings of laws of thought,

do you think such laws exist?
Can we linger on that?
What's your intuition here?
Oh, definitely.
I mean, the fact is, look, the logic is but the first step.
There are many other kinds of calculi
about things that we consider,
about sort of things that happen in the world
or things that are meaningful.
Well, how do you know logic is not the last step?
You know what I mean?
Well, because we can plainly see that their thing...
I mean, if you say, here's a sentence
that is syntactically correct, okay?
You look at it and you're like,
the happy electron, you know, eight,
I don't know what, something that it just...
You look at it and it's like, this is meaningless.
It's just a bunch of words.
It's syntactically correct.
The nouns and the verbs are in the right place,
but it just doesn't mean anything.
And so, there clearly is some rule
that there are rules that determine
when a sentence has the potential to be meaningful
that go beyond the pure parts of speech syntax.
And so, the question is, what are those rules
and are there a fairly finite set of those rules?
My guess is that there's a fairly finite set of those rules.
And they, you know, once you have those rules,
you have a kind of a construction kit,
just like the rules of syntactic grammar
give you a construction kit
for making syntactically correct sentences.
So, you can also have a construction kit
for making semantically correct sentences.
Those sentences may not be realized in the world.
I mean, I think, you know, the elephant flew to the moon.
A syntactically, a semantically, you know,
we know we have an idea.
If I say that to you, you kind of know what that means.
But the fact is it hasn't been realized
in the world, so to speak.

So, semantically correct perhaps
as things that can be imagined with the human mind, no.
Things that are consistent with both our imagination
and our understanding of physical reality.
Yeah, good question.
I mean, it's a good question.
It's a good question.
I mean, I think it is given the way we have constructed language,
it is things which fit
with the things we're describing in language.
It's a bit circular in the end,
because, you know, you can,
and the sort of boundaries of what is physically realizable.
Okay, let's take the example of motion, okay?
Motion is a complicated concept.
It might seem like it's a concept
that should have been figured out by the Greeks, you know, long ago.
But it's actually a really pretty complicated concept,
because what is motion?
Motion is you can go from place A to place B,
and it's still you when you get to the other end, right?
You take an object, you move it,
and it's still the same object,
but it's in a different place.
Now, even in ordinary physics,
that doesn't always work that way.
If you're near a space-time singularity in a black hole,
for example, and you take your teapot or something,
you don't have much of a teapot by the time
it's near the space-time singularity.
It's been completely, you know, deformed beyond recognition.
But so that's a case where pure motion doesn't really work.
You can't have a thing stay the same.
But so this idea of motion is something
that sort of is a slightly complicated idea.
But once you have the idea of motion,
you can start, once you have the idea
that you're going to describe things as being
the same thing, but in a different place.
That sort of abstracted idea then has, you know,
that has all sorts of consequences,
like this transitivity of motion,
go from A to B, B to C, you've gone from A to C.

And that's, so that level of description,
you can have what are sort of inevitable consequences.
They're inevitable features of the way
you've sort of set things up.
And that's, I think, what this sort of semantic grammar
is capturing is things like that.
And I, you know, I think that it's a question
of what does the word mean when you say I go from,
I move from here to there.
Well, it's complicated to say what that means.
This is this whole issue of, you know,
is pure motion possible, et cetera, et cetera, et cetera.
But once you have kind of got an idea of what that means,
then there are inevitable consequences of that idea.
But the very idea of meaning,
it seems like there's some words that become,
it's like there's a latent ambiguity to them.
I mean, it's the word like emotionally loaded words,
like hate and love.
Right.
It's like, what are they, what do they mean exactly?
So especially when you have relationships
between complicated objects,
we seem to take this kind of shortcut,
descriptive shortcut to describe like,
object A hates object B.
What's that really mean?
Right.
Well, words are defined by kind of our social use of them.
I mean, it's not, you know, a word in computational language,
for example, when we say we have a construct there,
we expect that that construct is a building block
from which we can construct an arbitrarily tall tower.
So we have to have a very solid building block.
And, you know, we have to, it turns into a piece of code,
it has documentation, it's, you know, it's a whole thing.
But the word hate, you know, the documentation for that word,
well, there isn't a standard documentation
for that word, so to speak.
It's a complicated thing defined by kind of how we use it.
When, you know, if it wasn't for the fact
that we were using language, I mean, so,
so what is language at some level?

Language is a way of packaging thoughts
so that we can communicate them to another mind.
Can these complicated words be converted
into something that a computation engine can use?
Right, so I think the answer to that is that
what one can do in computational language
is define, make a specific definition.
And if you have a complicated word,
like let's say the word eat, okay?
You'd think that's a simple word, it's, you know,
animals eat things, whatever else.
But, you know, you do programming, you say,
this function eats arguments.
Which is sort of poetically similar
to the animal eating things.
But if you start to say, well, what are the implications
of, you know, the function eating something?
You know, can the function be poisoned?
Well, maybe it can, actually.
But, you know, if there's a type mismatch
or something in some language.
But, you know, in what, how far does that analogy go?
And it's just an analogy.
Whereas if you use the word eat
in a computational language level,
you would define there isn't a thing
which you anchor to the kind of natural language
concept eat.
But it is now some precise definition of that
that then you can compute things from.
But don't you think the analogy is also precise?
Software eats the world?
Don't you think there's a,
there's something concrete
in terms of meaning about analogies?
Sure.
But the thing that sort of is the first target
for computational language is to take sort of
the ordinary meaning of things
and try and make it precise.
Make it sufficiently precise.
You can build these towers of computation on top of it.
So it's kind of like if you start with a piece of poetry

and you say, I'm going to define my program
with this piece of poetry.
It's kind of like that's,
that's a difficult thing.
It's better to say, I'm going to just have this
boring piece of prose and it's using words
in the ordinary way.
It's time communicating with my computer
and that time going to build the solid building block
from which I can construct this whole kind of
computational tower.
So there's some sense where if you take a poem
and reduce it to something computable,
you're going to have very few things left.
So maybe there's a bunch of human interaction
that's just poetic,
aimless nonsense.
That's just like a recreational,
like hamster in a wheel.
Well, I think that that's a complicated thing
because in a sense, human linguistic communication is
there's one mind.
It's producing language.
That language is having an effect on another mind.
And the question of,
there's sort of a type of effect
that is well-defined, let's say,
where, for example, it's very independent of the two minds,
that there's communication where it can matter a lot,
sort of what the experience of one mind is
versus another one and so on.
Yeah, but what is the purpose
of natural language communication?
I think the...
Versus, so computation,
computational language somehow feels more amenable
to the definition of purpose.
It's like, yeah, you're given two clean representations
of a concept and you can build a tower based on that.
Is natural language the same thing but more fuzzy?
Well, I think the story of natural language,
that's the great invention of our species.
We don't know whether it exists in other species,

but we know it exists in our species.
It's the thing that allows you
to sort of communicate abstractly
from like one generation of the species to another.
There is an abstract version of knowledge
that can be passed down.
It doesn't have to be genetics.
It doesn't have to be...
You don't have to apprentice the next generation of birds
to the previous one to show them how something works.
There is this abstracted version of knowledge
that can be passed down.
Now, that it relies on...
It still tends to rely because language is fuzzy.
It does tend to rely on the fact that if we look at
some ancient language where we don't have a chain
of translations from it until what we have today,
we may not understand that ancient language.
And we may not understand its concepts
may be different from the ones that we have today.
We still have to have something of a chain,
but it is something where we can realistically expect
to communicate abstract ideas,
and that's one of the big roles of language.
I think in...
That's been this ability to
concretify abstract things
is what language has provided.
Do you see natural language and thought as the same?
The stuff that's going inside your mind?
Well, that's been a long debate in philosophy.
It seems to become more important now
when we think about how intelligent GPT is.
Whatever that means.
Whatever that means,
but it seems like the stuff that's going on in the human mind
seems something like intelligence in its language.
But we call it intelligence.
Yes.
And so you start to think,
okay, what's the relationship between thought,
the language of thought, the laws of thought,
the laws of the words like reasoning,

and the laws of language,
and how that has to do with computation,
seems like more rigorous, precise ways of reasoning.
Right.
Which are beyond human.
I mean, much of what computers do, humans do not do.
I mean, you might say...
Humans are a subset, presumably.
Yes.
Hopefully.
Yes.
Yes, right.
You know, you might say,
who needs computation when we have large language models?
Large language models can just, you know,
eventually you'll have a big enough neural net
it can do anything.
But they're really doing the kinds of things
that humans quickly do.
And there are plenty of sort of formal things
that humans never quickly do.
For example, I don't know, you know,
people can do mental arithmetic.
They can do a certain amount of math in their minds.
I don't think many people can run a program
in their minds of any sophistication.
It's just not something people do.
It's not something people have even thought of doing
because it's kind of not, you know,
you can easily run it on a computer.
An arbitrary program.
Yeah.
Are we running specialized programs?
Yeah, yeah, yeah.
But if I say to you, here's a Turing machine.
Yeah.
You know, tell me what it does after 50 steps.
And you're like trying to think about that in your mind.
That's really hard to do.
It's not what people do.
I mean, it's...
Well, in some sense, people program,
they build a computer, they program it,

just to answer your question about
what the system does after 50 steps.
I mean, humans build computers.
Yes.
Yes.
But they've created something which is then,
you know, then when they run it,
it's doing something different than
what's happening in their minds.
I mean, they've outsourced that piece of computation
from something that is internally happening in their minds
to something that is now a tool that's external to their mind.
So, by the way, humans, to you, didn't invent computers.
They discovered them.
They discovered computation.
Which...
They invented the technology of computers.
The computer is just the kind of way
to plug into this whole stream of computation.
Yes.
There's probably other ways.
There's probably a lot of ways.
Well, for sure.
I mean, the, you know, the particular ways
that we make computers out of semiconductors
and electronics and so on,
that's the particular technology stack we built.
I mean, the story of a lot of what people try to do
is finding different sort of underlying physical,
you know, infrastructure for doing computation.
You know, biology does lots of computation.
It does it using an infrastructure that's different
from semiconductors and electronics.
It's a, you know, it's a molecular scale,
sort of computational process
that hopefully will understand more about.
I have some ideas about understanding more about that.
But, you know, that's another, you know,
it's another representation of computation.
Things that happen in the physical universe
at the level of, you know, these evolving hypergraphs and so on.
That's another sort of implementation layer
for this abstract idea of computation.

So if GPT or large language models are starting to form,
starting to develop or implicitly understand the laws
of language and thought,
do you think they can be made explicit?
Yes.
How?
With a bunch of effort.
I mean, it's like doing natural science.
I mean, what is happening in natural science?
You have the world that's doing all these complicated things
and then you discover, you know, Newton's laws, for example.
This is how motion works.
This is the way that this particular sort of idealization
of the world, this is how we describe it
in a simple computationally reducible way.
And I think it's the same thing here.
It's there are sort of computationally reducible aspects
of what's happening that you can get a kind of narrative theory for
just as we've got narrative theories and physics and so on.
Do you think it will be depressing or exciting
when all the laws of thought are made explicit,
human thought are made explicit?
I think that once you understand computational reducibility,
it is, it's neither of those things
because the fact is people say, for example,
people will say, oh, but, you know, I have free will.
I kind of, you know, I operate in a way that is,
you know, they have the idea that they're doing something
that is sort of internal to them
that they're figuring out what's happening.
But in fact, we think there are laws of physics
that ultimately determine, you know, every nerve,
every electrical impulse and a nerve and things like this.
So you might say, isn't it depressing
that we are ultimately just determined by the rules of physics,
so to speak?
It's the same thing. It's at a higher level.
It's like it's a shorter distance to get from kind of semantic grammar
to the way that we might construct a piece of text
than it is to get from individual nerve firings
to how we construct a piece of text.
But it's not fundamentally different.
And by the way, as soon as we have this kind of level of,

you know, this other level of description,
it's kind of, it helps us to go even further.
So we'll end up being able to produce more and more complicated
kinds of things that just like when we, you know,
if we didn't have a computer and we knew certain rules,
we could write them down and go a certain distance.
But once we have a computer, we can go vastly further
and this is the same kind of thing.
You wrote a blog post titled,
What is Chad GPT doing and why does it work?
We've been talking about this,
but can we just step back and linger on this question?
What's Chad GPT doing?
What are these a bunch of billion parameters
trained on a large number of words?
Why does it seem to work again?
Is it because of the point you made
that there's laws of language
that can be discovered by such a process?
Is there something more to it?
Let's talk about sort of the low level
of what Chad GPT is doing.
I mean, ultimately, you give it a prompt,
it's trying to work out, you know,
what should the next word be, right?
Which is wild.
Isn't that, isn't that surprising to you
that this kind of low level, dumb training procedure
can create something syntactically correct first
and then semantically correct second?
You know, the thing that has been sort of a story of my life
is realizing that simple rules
can do much more complicated things than you imagine.
That something that starts simple
and starts simple to describe
can grow a thing that is, you know,
vastly more complicated than you can imagine.
And honestly, it's taken me,
I don't know, I've sort of been thinking about this
now 40 years or so, and it always surprises me.
I mean, even for example, in our physics project,
sort of thinking about the whole universe
growing from these simple rules,

it will resist because I keep on thinking,
you know, how can something really complicated
arise from something that simple?
It just seems, you know, it seems wrong,
but yet, you know, the majority of my life
I've kind of known from things I've studied
that this is the way things work.
So yes, it is wild that it's possible
to write a word at a time
and produce a coherent essay, for example.
But it's worth understanding kind of how that's working.
I mean, it's kind of like if it was going to say,
you know, the cat sat on thee,
what's the next word?
Okay, so how does it figure out the next word?
Well, it's seen a trillion words written on the internet.
And it's seen the cat sat on the floor,
the cat sat on the sofa,
the cat sat on the whatever.
So its minimal thing to do is just say,
let's look at what we saw on the internet.
We saw, you know, 10,000 examples of the cat sat on thee.
What was the most probable next word?
Let's just pick that out and say that's the next word.
And that's kind of what at some level is trying to do.
Now, the problem is there isn't enough text on the internet
to, if you have a reasonable length of prompt,
that specific prompt will never have occurred on the internet.
And as you kind of go further,
there just won't be a place where you could have trained,
you know, where you could just worked out probabilities
from what was already there.
You know, like if you say two plus two,
there'll be a zillion examples of two plus two
equaling four and a very small number of examples
of two plus two equals five and so on.
And you can pretty much know what's going to happen.
So then the question is, well,
if you can't just work out from examples what's going to happen,
just no probabilistic for you from examples
what's going to happen, you have to have a model.
And this kind of an idea,
this idea of making models of things

is an idea that really, I don't know,
I think Galileo probably was one of the first people
who sort of worked this out.
I mean, it's kind of like, you know,
I think I gave an example of that little book I wrote
about about chat GBT where it's kind of like, you know,
Galileo was dropping cannonballs
off the different floors of the Tower of Pisa.
And it's like, okay, you drop a cannonball off this floor,
you drop a cannonball off this floor,
you miss floor five or something for whatever reason.
But you know the time it took the cannonball
to fall to the ground from floors one, two, three, four,
six, seven, eight, for example,
then the question is, can you work out?
Can you make a model which figures out
how long would it take the ball to fall to the ground
from the floor you didn't explicitly measure?
And the thing Galileo realized
is that you can use math,
you can use mathematical formulas to make a model
for how long it will take the ball to fall.
So now the question is, well, okay,
you want to make a model for, for example,
something much more elaborate
like you've got this arrangement of pixels
and is this arrangement of pixels an A or a B?
Does it correspond to something we'd recognize as an A or a B?
And you can make a similar kind, you know,
each pixel is like a parameter in some equation
and you could write down this giant equation
where the answer is either, you know, A or, you know,
one or two A or B.
And the question is then,
what kind of a model successfully reproduces
the way that we humans would,
would conclude that this is an A and this is a B?
You know, if there's a complicated extra tail
on the top of the A,
would we then conclude something different?
What is the type of model that maps well
into the way that we humans make distinctions about things?
And the big kind of meta discovery

is neural nets are such a model.
It's not obvious they would be such a model.
It could be that human distinctions are not captured.
You know, we could try searching around for a type of model
that could be a mathematical model.
It could be some model based on something else
that captures kind of typical human distinctions about things.
It turns out this model that actually is very much
the way that we think the architecture of brains works
that perhaps not surprisingly,
that model actually corresponds
to the way we make these distinctions.
And so, you know, the core next point is that
the kind of model, this neural net model,
makes sort of distinctions and generalizes things
in sort of the same way that we humans do it.
And that's why when you say, you know,
the cat sat on the green blank,
even though it never, it didn't see many examples
of the cat sat on the green whatever,
it can make a, or the odd VARC sat on the green whatever.
I'm sure that particular sentence
does not occur on the internet.
And so it has to make a model that concludes what,
you know, it has to kind of generalize
from the actual examples that it's seen.
And so, you know, that's the fact is that neural nets
generalize in the same kind of way that we humans do.
If we were, you know, the aliens might look
at our neural net generalizations and say,
that's crazy, you know, that thing,
when you put that extra little dot on the A,
that isn't an A anymore, that's, you know,
that messed the whole thing up.
But for us humans, we make distinctions
which seem to correspond to the kinds of distinctions
that neural nets make.
So then, you know, the thing that is just amazing
to me about chat GBT is how similar
the structure it has is to the very original way
people imagine neural nets might work back in 1943.
And, you know, there's a lot of detailed engineering,
you know, great cleverness,

but it's really the same idea.
And in fact, even the sort of elaborations
of that idea where people said,
let's put in some actual particular structure
to try and make the neural net more elaborate
to be very clever about it,
most of that didn't matter.
I mean, there's some things that seem to, you know,
when you train this neural net, you know,
the one thing this kind of transformer architecture,
this attention idea, that really has to do with,
does every one of these neurons connect
to every other neuron, or is it somehow
causally localized, so to speak?
Does it like we're making a sequence of words
and the words depend on previous words,
rather than just everything can depend on everything?
And that seems to be important in just organizing things
so that you don't have a sort of a giant mess.
But the thing, you know, the thing worth understanding
about what is chat GPT in the end?
I mean, what is a neural net in the end?
A neural net in the end is each neuron has a,
it's taking inputs from a bunch of other neurons.
It's eventually, it's going to have,
it's going to have a numerical value,
it's going to compute some number,
and it's saying, I'm going to look at the neurons above me,
it's kind of a series of layers,
it's going to look at the neurons above me,
and it's going to say, what are the values of all those neurons?
Then it's going to add those up and multiply them by these weights,
and then it's going to apply some function that says,
if it's bigger than zero or something,
then make it one, or otherwise make it zero,
or some slightly more complicated function,
you know very well how this works.
It's a giant equation with a lot of variables.
And figuring out where the ball falls
when you don't have data on the fourth floor,
this, the equation here is not as simple as the equation.
Right, it's an equation with 175 billion terms.
And it's quite surprising that in some sense,

a simple procedure of training such an equation
can lead to a good representation of natural language.
Right, the real issue is, you know,
this architecture of a neural net where what's happening is,
you know, you've turned,
so neural nets always just deal with numbers.
And so, you know, you've turned the sentence
that you started with into a bunch of numbers,
like let's say by mapping, you know,
each word of the 50,000 words in English,
you just map each word or each part of a word into some number.
You feed all those numbers in,
and then the thing is going to,
and then those numbers just go into the values of these neurons,
and then what happens is it's just rippling down,
going layer to layer until it gets to the end.
I think chat UBG has about 400 layers,
and you're just, you know, it just goes once through.
It just, every new word it's going to compute just says,
here are the numbers from the words before.
Let's compute the, what does it compute?
It computes the probabilities that it estimates
for each of the possible 50,000 words that could come next.
And then it decides sometimes it will use the most probable word,
sometimes it will use not the most probable word.
It's an interesting fact that there's this so-called temperature parameter,
which, you know, at temperature zero,
it's always using the most probable word that it can,
that it estimated was the most probable thing to come next.
You know, if you increase the temperature,
it'll be more and more kind of random in its selection of words.
It'll go down to lower and lower probability words.
The thing I was just playing with actually recently
was the transition that happens as you increase the temperature.
The thing goes bonkers.
At a particular, you know, sometimes at a particular temperature,
maybe about 1.2 is the thing I was noticing from yesterday, actually,
that, you know, usually it's giving reasonable answers
and then at that temperature with some probability,
it just starts spouting nonsense.
And, you know, nobody knows why this happens.
I mean, it's, and by the way, I mean,
the thing to understand is it's putting down one word at a time,

but the outer loop of the fact that it says,
okay, I put down a word.
Now let's take the whole thing I wrote so far.
Let's feed that back in.
Let's put down another word.
That outer loop, which seems almost trivial,
is really important to the operation of the thing.
And for example, one of the things that is kind of funky
is it'll give an answer and you say to it,
is that answer correct?
And it'll say no.
And why is that happening?
It's fascinating, right?
Right. Why can it do that?
Well, the answer is because it is going one word at a time
sort of forwards and it didn't, you know,
it came along with some sort of chain of thought in a sense
and it came up with completely the wrong answer.
But as soon as you feed it,
the whole thing that it came up with,
it immediately knows that that isn't right.
It immediately can recognize that was a, you know,
a bad syllogism or something and can see what happened,
even though as it was being led down this garden path,
so to speak, it didn't, it came to the wrong place.
But it's fascinating that this kind of procedure converges
to something that forms a pretty good compressed representation
of language on the Internet.
That's quite...
Right, right, right.
I'm not sure what to make of it.
Well, look, I think, you know,
there are many things we don't understand, okay?
So for example, you know, 175 billion weights,
it's maybe about a trillion bytes of information.
Which is very comparable to the training set that was used.
And, you know, why that, why kind of,
it sort of stands to some kind of reason
that the number of weights in the neural net,
I don't know, I can't really argue that.
I can't really give you a good, you know,
in a sense, the very fact that, you know,
insofar as there are definite rules of what's going on,

you might expect that eventually
we'll have a much smaller neural net
that will successfully capture what's happening.
I don't think the best way to do it is probably a neural net.
I think a neural net is what you do when you don't know
any other way to structure the thing.
And it's a very good thing to do
if you don't know any other way to structure the thing.
And for the last 2,000 years,
we haven't known any other way to structure it.
So this is a pretty good way to start.
But that doesn't mean you can't find,
sort of in a sense, more symbolic rules for what's going on,
that, you know, much of which will then be,
you can kind of get rid of much of the structure
of the neural net and replace it by things
which are sort of pure steps of computation,
so to speak, sort of with neural net stuff around the edges.
And that becomes just a, you know,
it's just a much simpler way to do it.
So the neural net, you hope will reveal
to us good symbolic rules
that make the need of the neural net less and less and less.
Right.
And there will still be some stuff that's kind of fuzzy,
just like, you know, there are things that,
it's like this question of what can we formalize?
What can we turn into computational language?
What is just sort of, oh, it happens that way
just because brains are set up that way.
What do you think are the limitations
of large language models, just to make it explicit?
Well, I mean, I think that deep computation
is not what large language models do.
I mean, that's just, it's a different kind of thing.
You know, the outer loop of a large language model,
if you're trying to do many steps in a computation,
the only way you get to do that right now
is by spooling out, you know,
all the whole chain of thought as a bunch of words, basically.
And, you know, you can make a Turing machine out of that
if you want to, I just was doing that construction.
You know, in principle, you can make an arbitrary computation

by just spooling out the words,
but it's a bizarre and inefficient way to do it.
But it's something where the, you know,
I think that's, you know, sort of the deep computation.
It's really what humans can do quickly.
Large language models will probably be able to do well.
Anything that you can do kind of off the top of your head type thing
is really, you know, is good for large language models.
And the things you do off the top of your head,
you may not get them always right,
but, you know, it's thinking through the same way we do.
But I wonder if there's an automated way to do something
that humans do well, much faster to where it like loops,
so generate arbitrary large code bases
of Wolfram language, for example.
Well, the question is, what do you want the code base to do?
Escape, control, and take over the world.
Okay. So, you know, the thing is, when people say,
you know, we want to build this giant thing, right?
A giant piece of computational language.
In a sense, it's sort of a failure of computational language.
If the thing you have to build, in other words,
if we have a description, if you have a small description,
that's the thing that you represent in computational language.
And then the computer can compute from that.
Yes.
So, in a sense, in, you know,
when as soon as you're giving a description,
that, you know, if you have to somehow make that description
and something, you know, definite, something formal,
and once, and to say, to say,
okay, I'm going to give this piece of natural language,
and then it's going to splurt out this giant formal structure,
that, in a sense, that doesn't really make sense,
because except insofar as that piece of natural language
kind of plugs into what we socially know, so to speak,
plugs into kind of our corpus of knowledge,
then, you know, that's the way we're capturing a piece
of that corpus of knowledge,
but hopefully we will have done that in computational language.
How do you make it do something that's big?
Well, you know, you have to have a way to describe what you want.
I can make it more explicit if you want.

How about I just pop into my head.
Iterate through all the members of Congress
and figure out how to convince them that they have to
let me, meaning the system, become president,
pass all the laws that allows AI systems
to take control and be the president, I don't know.
So that's a very explicit, like,
figure out the individual life story
of each congressman, each senator, anybody,
I don't know what's required to really kind of pass legislation,
and figure out how to control them and manipulate them,
get all the information.
What would be the biggest fear of this congressman
and in such a way that you can take action on it
in the digital space?
So maybe threaten the destruction reputation
or something like this.
Right.
If I can describe what I want, you know,
to what extent can a large language model automate that?
With the help of the concretization
of something like Wolfram Language,
that makes it more, yeah, grounded.
I think it can go rather a long way.
I'm also surprised how quickly I was able to generate.
Yeah, yeah, right.
An attack.
That's a, yeah, you know.
I swear, I swear I did not think about this before,
and it's funny how quickly, which is a very concerning thing,
because that probably this idea will probably do
quite a bit of damage, and there might be
a very large number of other such ideas.
Well, I'll give you a much more benign version of that idea.
Okay?
You're going to make an AI tutoring system.
And, you know, that's a benign version of what you're saying,
is I want this person to understand this point.
Yes.
You know, you're essentially doing machine learning
where the, you know, the loss function,
the thing you're trying to get to is
get the human to understand this point.

And when you do a test on the human,
that they, yes, they correctly understand
how this or that works.
And I am confident that, you know,
sort of a large language model type technology
combined with computational language
is going to be able to do pretty well
at teaching us humans things.
And it's going to be an interesting phenomenon
because, you know, sort of individualized teaching
is a thing that has been kind of a, you know,
a goal for a long time.
I think we're going to get that.
And I think more, you know, it has many consequences
for, you know, like, just, you know,
if you know me as an, if you, the AI know me,
tell me, I'm about to do this thing.
What is the, what are the three things I need to know?
You know, given what I already know,
you know, what's, let's say,
I'm looking at some paper or something, right?
It's like, there's a version of the summary of that paper
that is optimized for me, so to speak.
Where it really is, and I think that's really going to work.
It could understand the major gaps in your knowledge
that it filled would actually give you
a deeper understanding of the topic here.
Right.
And that's a, you know, that's an important thing
because it really changes, actually, I think, you know,
when you think about education and so on,
it really changes kind of what's worth doing,
what's not worth doing and so on.
It makes, you know, I know in my life
I've learned lots of different fields.
And, you know, so I don't know,
I have every time I'm always think this is the one
that's going to, I'm not going to be able to learn.
Yeah.
But turns out, sort of, there are sort of
meta methods for learning these things in the end.
And, you know, I think this idea that it becomes easier to,
you know, it becomes easier to be fed knowledge,

so to speak.
And it becomes, you know,
if you need to know this particular thing,
you can, you know, you can get taught it in an efficient way.
It's something I think is sort of an interesting feature.
And I think it makes the, you know,
things like the value of big towers of specialized knowledge
become less significant compared to the kind of meta knowledge
of sort of understanding kind of the big picture
and being able to connect things together.
I think that, you know, there's been this huge trend
of let's be more and more specialized
because we have to, you know,
we have to sort of ascend these towers of knowledge.
But by the time you can get, you know,
more automation of being able to get to that place on the tower
without having to go through all those steps,
I think it sort of changes that picture.
Interesting.
So your intuition is that in terms of the collective intelligence
of the species and the individual minds
that make up that collective,
there'll be more, there will trend towards being generalists
and being kind of philosophers.
That's what I think.
I think that's where the humans are going to be useful.
I think that a lot of these kind of,
the drilling, the mechanical working out of things
is much more automatable.
It's much more AI territory, so to speak.
No more PhDs.
Well, that's interesting.
Yes, I mean, you know, the kind of the specialization,
this kind of tower of specialization,
which has been a feature of, you know,
we've accumulated lots of knowledge in our species
and, you know, in a sense, every time we have
a kind of automation, a building of tools,
it becomes less necessary to know that whole tower
and it becomes something where you can just use a tool
to get to the top of that tower.
I think that, you know, the thing that is ultimately,
you know, when we think about, okay,

what do the AIs do versus what do the humans do?
It's like AIs, you tell them, you say,
go achieve this particular objective.
Okay, they can maybe figure out a way to achieve that objective.
We say, what objective would you like to achieve?
The AI has no intrinsic idea of that.
It's not a defined thing.
That's a thing which has to come from some other,
you know, some other entity.
And insofar as we are in charge, so to speak,
whatever it is, and our kind of web of society
and history and so on is the thing that is defining
what objective we want to go to.
That's, you know, that's a thing that we humans
are necessarily involved in, so to say.
To push back a little bit, don't you think that GPT,
feature versions of GPT would be able to give a good answer
to what objective would you like to achieve?
From what basis?
If they say, look, here's the terrible thing
that could happen, okay?
They're taking the average of the internet
and they're saying, you know,
from the average of the internet, what do people want to do?
Well, that's the almost scatting of
the most entertaining outcome is the most likely.
Okay, I haven't heard that one from him yet.
That could be one objective,
is maximize global entertainment,
the darker version of that is drama,
the good version of that is fun.
Right, so I mean, this question of what,
you know, if you say to the AI, you know,
what does the species want to achieve?
Yes, okay.
There'll be an answer, right?
There'll be an answer.
It'll be what the average of the internet
says the species wants to achieve.
Well, I think you're using the word average
very loosely there, right?
So I think the answers will become
more and more interesting as these language models

are trained better and better.
No, but I mean, in the end,
it's a reflection back of what we've already said.
Yes, but there's a deeper wisdom
to the collective intelligence,
presumably than each individual.
Maybe.
Isn't that what we're trying to do as a society?
Well, I mean, that's an interesting question.
I mean, you know, insofar as some of us,
you know, work on trying to innovate
and figure out new things and so on,
it is sometimes, it's a complicated interplay
between sort of the individual doing the crazy thing
off in some spur, so to speak,
versus the collective that's trying to do
sort of the high inertia average thing.
And it's, you know, sometimes the collective,
you know, is bubbling up things that are interesting
and sometimes it's pulling down
kind of the attempt to make this kind of innovative direction.
What don't you think the large language models
would see beyond that simplification
will say maybe intellectual and career diversity
is really important, so you need the crazy people
on the outlier, on the outskirts,
and so like the actual,
what's the purpose of this whole thing
is to explore through this kind of
dynamics that we've been using as a human civilization
which is most of us focus on one thing
and then there's the crazy people on the outskirts
doing the opposite of that one thing
and you kind of, they pull the whole society together,
there's the mainstream science
and then there's the crazy science
and that's just been enough of the history of human civilization
and maybe the AI system will be able to see that
and the more and more impressed we are by
a language model telling us this,
the more control we'll give it to it
and the more we'll be willing to let it run our society
and hence there's this kind of loop

where the society could be manipulated
to let the AI system run it.
Right, well I mean, look, one of the things
that's sort of interesting is
we might say we always think we're making progress
but yet if, you know, in a sense by saying
let's take what already exists
and use that as a model for what should exist,
then, you know, it's interesting that for example,
you know, many religions have taken that point of view.
There is a, you know, a sacred book that got written at Timex
and it defines how people should act for all future time
and that's, you know, it's a model that people have operated with
and in a sense, you know, this is a version
of that kind of statement.
It's like take the 2023 version of sort of how the world
has exposed itself and use that to define
what the world should do in the future.
But it's not, it's an imprecise definition, right?
Because just like with religious texts and with GPT,
the human interpretation of what GPT says
will be the, will be the perturbation in the system.
It'll be the noise, it'd be full of uncertainty.
It's not like GPT, Chad GPT will tell you exactly what to do.
It'll tell you a proxy, a narrative of what, like, you know,
it's like turn the other cheek kind of narrative, right?
That's not a fully instructive narrative.
Well, until the AIs control all the systems in the world.
They will be able to very precisely tell you what to do.
Well, they'll do what they, you know, they'll just do this or that thing.
And not only that, they'll be auto-suggesting to each person,
you know, do this next, do that next.
So I think it's a, it's a slightly more prescriptive situation
than one has typically seen.
But, you know, I think this, this whole question of sort of what,
what's left for the humans, so to speak, to what extent do we, you know,
this idea that there is an existing kind of corpus of purpose
for humans defined by what's on the internet and so on,
that's an important thing.
But then the question of sort of, as we explore what we can think of
as the computational universe, as we explore all these different possibilities
for what we could do, all these different inventions we could make,
all these different things, the question is, which ones do we choose to follow?

Those choices are the things that, in a sense, if the humans want to still have kind of human progress, that's what we, we get to make those choices, so to speak.
In other words, there's this idea, if you say, let's take the kind of what exists today and use that as the determiner of all of what there is in the future.
The thing that is sort of the opportunity for humans is there will be many possibilities thrown up.
There are many different things that could happen or be done.
And the, in so far as we want to be in the loop, the thing that makes sense for us to be in the loop doing is picking which of those possibilities we want.
But the degree to which there's a feedback loop of the idea that we're picking something starts becoming questionable because we're influenced by the very systems.
Absolutely.
If that becomes more and more source of our education and wisdom and knowledge.
Absolutely.
Right.
The AIs take over.
I mean, I've thought for a long time that it's the AR auto-suggestion.
That's really the thing that makes the AIs take over.
It's just that the humans just follow, you know.
Yeah.
We will no longer write emails to each other.
We'll just send the auto-suggested email.
Yeah.
Yeah.
But the thing where humans are potentially in the loop is when there's a choice and when there's a choice which we could make based on our kind of whole web of history and so on.
Yeah.
That's in so far as it's all just determined the humans don't have a place.
And by the way, I mean, at some level, it's all kind of a complicated philosophical issue because at some level the universe is just doing what it does.
We are parts of that universe that are necessarily doing what we do, so to speak.
And yet we feel we have sort of agency in what we're doing and that's its own separate kind of interesting issue.
And we also kind of feel like we're the final destination of what the universe was meant to create, but we very well could be and likely are some kind of intermediate step, obviously.
Yeah.

Well, we're most certainly some intermediate step.
The question is if there's some cooler, more complex, more interesting thing that's going to materialize.
The computational universe is full of such things.
But in our particular pocket specifically, if this is the best we're going to do or not, that's kind of a...
We can make all kinds of interesting things in the computational universe.
When we look at them, we say, yeah, that's a thing.
It doesn't really connect with our current way of thinking about things.
It's like in mathematics.
We've got certain theorems.
There are about three or four million that human mathematicians have written down and published and so on.
But they're an infinite number of possible mathematical theorems.
We just go out into the universe of possible theorems and pick another theorem.
And then people will say, well, they look at it and they say, I don't know what this theorem means.
It's not connected to the things that are part of kind of the web of history that we're dealing with.
I think one point to make about sort of understanding AI and its relationship to us is as we have this kind of whole infrastructure of AIs doing their thing and doing their thing in a way that is perhaps not readily understandable by us humans, you might say that's a very weird situation.
How come we have built this thing that behaves in a way that we can't understand that's full of computational irreducibility, et cetera, et cetera, et cetera?
What is this?
What's it going to feel like when the world is run by AIs whose operations we can't understand?
And the thing one realizes is actually we've seen this before.
That's what happens when we exist in the natural world.
The natural world is full of things that operate according to definite rules.
They have all kinds of computational irreducibility.
We don't understand what the natural world is doing.
Occasionally, when you say, are the AIs going to wipe us out, for example?
Well, it's kind of like, is the machination of the AIs going to lead to this thing that eventually comes and destroys the species?
Well, we can also ask the same thing about the natural world.
The machination of the natural world going to eventually lead to this thing that's going to make the earth explode or something like this.
Those are questions.
And insofar as we think we understand what's happening in the natural world, that's a result of science and natural science and so on.
One of the things we can expect when there's this giant infrastructure of the AIs

is that's where we have to kind of invent a new kind of natural science
that kind of is the natural science that explains to us how the AIs work.
It's kind of like we have a, I don't know, a horse or something
and we're trying to ride the horse and go from here to there.
We don't really understand how the horse works inside,
but we can get certain rules and certain approaches that we take
to persuade the horse to go from here to there and take us there.
And that's the same type of thing that we're kind of dealing with
with the sort of incomprehensible, computationally irreducible AIs,
but we can identify these kinds of,
we can find these kind of pockets of reducibility that we can kind of,
you know, we're grabbing onto the mane of the horse or something
to be able to ride it or we figure out, you know,
if we do this or that to ride the horse,
that that's a successful way to get it to do what we're interested in doing.
There does seem to be a difference between a horse
and a large language model
or something that could be called AGI connected to the internet.
So let me just ask you about big philosophical question about the threats of these things.
There's a lot of people like Eliezer Edkowski who worry about
the existential risks of AI systems.
Is that something that you worry about?
You know, sometimes when you're building an incredible system like Wolfram Alpha,
you can kind of get lost in it.
Oh, I try and think a little bit about the implications of what one's doing.
You know, it's like the Manhattan Project kind of situation
where you're like, it's some of the most incredible physics and engineering being done,
but it's like, huh, where's this going to go?
I think some of these arguments about kind of, you know,
there'll always be a smarter AI, there'll always be, you know,
and eventually the AIs will get smarter than us
and then all sorts of terrible things will happen.
To me, some of those arguments remind me of kind of the ontological arguments
for the existence of God and things like this.
They're kind of arguments that are based on some particular model,
fairly simple model often of kind of there is always a greater this, that and the other.
You know, this is, and that's, you know, those arguments,
what tends to happen in the sort of reality of how these things develop
is that it's more complicated than you expect,
that the kind of simple logical argument that says,
oh, eventually there'll be a superintelligence and then it will, you know, do this and that
turns out not to really be the story, it turns out to be a more complicated story.
So for example, here's an example of an issue.

Is there an apex intelligence?
Just like there might be an apex predator in some, you know, ecosystem.
Is there going to be an apex intelligence?
The most intelligent thing that there could possibly be, right?
I think the answer is no.
And in fact, we already know this and it's a kind of a back to the whole computational irreducibility story.
There's kind of a question of, you know, even if you have,
if you have sort of a Turing machine and you have a Turing machine
that runs as long as possible before it halts,
you say, is this the machine, is this the apex machine that does that?
There will always be a machine that can go longer.
And as you go out to the infinite collection of possible Turing machines,
you'll never have reached the end, so to speak.
You'll always be able to...
It's kind of like the same question of whether there'll always be another invention.
Will you always be able to invent another thing?
The answer is yes.
There's an infinite tower of possible inventions.
That's one definition of apex.
But the other is like, which I also think might be true.
Is there a species that's the apex intelligence right now on Earth?
So it's not trivial to say that humans are that?
Yeah, it's not trivial. I agree.
It's a, you know, I think one of the things that I've long been curious about,
kind of other intelligences, so to speak.
I mean, I, you know, I view intelligence as like computation.
And it's kind of a, you know, you're sort of, you have this set of rules,
you deduce what happens.
I have tended to think now that there's this sort of specialization of computation
that is sort of a consciousness-like thing
that has to do with these, you know, computational boundedness,
single thread of experience, these kinds of things
that are the specialization of computation
that corresponds to a somewhat human-like experience of the world.
Now the question is, so that's, you know, there may be other intelligences like, you know,
the aphorism, you know, the weather has a mind of its own.
It's a different kind of intelligence that can compute all kinds of things
that are hard for us to compute, but it is not well aligned with us,
with the way that we think about things.
It doesn't, it doesn't, it doesn't think the way we think about things.
And, you know, in this idea of different, different intelligences,
every different mind, every different human mind is a different intelligence

that thinks about things in different ways.
And, you know, in terms of the kind of formalism of our physics project,
we talk about this idea of a rural space,
the space of all possible sort of rural systems,
and different minds are in a sense at different points in rural space.
Human minds, ones that have grown up with the same kind of culture
and ideas and things like this, might be pretty close in rural space,
pretty easy for them to communicate, pretty easy to translate,
pretty easy to move from one place in rural space
that corresponds to one mind to another place in rural space
that corresponds to another sort of nearby mind.
When we deal with kind of more distant things in rural space,
like, you know, the pet cat or something,
you know, the pet cat has some aspects that are shared with us.
The emotional responses of the cat are somewhat similar to ours,
but the cat is further away in rural space than people are.
And so then the question is, you know, can we identify sort of the,
can we make a translation from our thought processes
to the thought processes of a cat or something like this?
And, you know, what will we get when we, you know,
what will happen when we get there?
And I think it's the case that many, you know, many animals,
I don't know, dogs, for example, you know,
they have elaborate olfactory systems,
they, you know, they have sort of the smell architecture
of the world, so to speak, in a way that we don't.
And so, you know, if you were sort of talking to the dog
and you could, you know, communicate in a language,
the dog will say, well, this is a, you know, a, you know,
a flowing, smelling, this, that, and the other thing,
concepts that we just don't have any idea about.
Now, what's interesting about that is,
one day we will have chemical sensors that do a really pretty good job,
you know, we'll have artificial noses that work pretty well
and we might have our augmented reality systems show us
kind of the same map that the dog could see and things like this,
you know, similar to what happens in the dog's brain.
And eventually, we will have kind of expanded in rural space
to the point where we will have those same sensory experiences
that dogs have, and we will have internalized
what it means to have, you know, the smell landscape or whatever.
And so then we will have kind of colonized that part of rural space
until, you know, we haven't gone, you know,

some things that, you know, animals and so on do,
we sort of successfully understand, others we do not.
And the question of what kind of, what is the, you know,
what representation, you know, how do we convert things
that animals think about to things that we can think about,
that's not a trivial thing.
And, you know, I've long been curious,
I had a very bizarre project at one point of trying to make
an iPad game that a cat could win against its owner.
Right, so it feels like there's a deep philosophical goal there, though.
Yes, yes.
I mean, you know, I was curious if, you know,
if pets can work in Minecraft or something,
and can construct things, what will they construct,
and will what they construct be something where we look at it,
and we say, oh yeah, I recognize that.
Or will it be something that looks to us
like something that's out there in the computational universe,
that one of my, you know, cellular automata might have produced
where we say, oh yeah, I can kind of see it,
operates according to some rules,
I don't know why you would use those rules,
I don't know why you would care.
Yeah, actually just to link on that seriously,
is there a connector in the real space between you and a cat
where the cat could legitimately win?
So iPad is a very limited interface.
I wonder if there's a game where cats win.
I think the problem is the cats don't tend to be that interested
in what's happening on the iPad.
Yeah, that's an interface issue, probably.
Yeah, right, right, right.
No, I think it is likely that, I mean, you know,
there are plenty of animals that would successfully eat us
if we were exposed to them.
And so there's, you know, it's going to pounce faster
than we can get out of the way and so on.
So there are plenty of, and probably it's going to,
you know, we think we've hidden ourselves,
but we haven't successfully hidden ourselves.
That's a physical strength.
I wonder if there's something
in more in the realm of intelligence

where an animal like a cat could out...
Well, I think there are things, certainly,
in terms of the speed of processing certain kinds of things,
for sure.
I mean, the question of what, you know,
is there a game of chess, for example,
is there cat chess that the cats could play against each other
and if we tried to play a cat, we'd always lose.
I don't know.
It might have to do with speed,
but it might have to do with concepts also.
There might be concepts in the cat's head.
I tend to think that our species,
from its invention of language,
has managed to build up this kind of tower of abstraction
that for things like a chess-like game,
will make us win.
In other words, we've become through the fact
that we've kind of experienced language
and learnt abstraction.
You know, we've sort of become smarter
at those kinds of abstract kinds of things.
Now, you know, that doesn't make us smarter
at catching a mouse or something.
It makes us smarter at the things that we've chosen
to sort of concern ourselves,
which are these kind of abstract things.
Yeah.
And I think, you know, this is, again,
back to the question of, you know, what does one care about?
You know, if one's the cat,
if you have the discussion with a cat,
if we can translate things to have the discussion with a cat,
the cat will say, you know,
I'm very excited that this light is moving
and will say, why do you care?
And the cat will say,
that's the most important thing in the world
that this thing moves around.
I mean, it's like when you ask about, I don't know,
you look at archaeological remains and you say,
these people had this, you know, belief system about this
and, you know, that was the most important thing

in the world to them.
And now we look at it and say,
we don't know what the point of it was.
I mean, I've been curious, you know,
there are these hand prints on caves
from 20,000 or more years ago.
And it's like, nobody knows what these hand prints were there for,
you know, that they may have been a representation
of the most important thing you can imagine.
They may just have been some, you know,
some kid who rubbed their hands in the mud
and stuck them on the walls of the cave.
You know, we don't know.
And I think, but this whole question of what, you know,
when you say this question of sort of,
what's the smartest thing around?
There's the question of,
what kind of computation are you trying to do?
If you're saying, you know, if you say,
you've got some well-defined computation
and how do you implement it?
Well, you could implement it by nerve cells, you know,
firing, you can implement it with silicon and electronics.
You can implement it by some kind of molecular computation
process in the human immune system
and some molecular biology kind of thing.
There are different ways to implement it.
And, you know, I think this question of sort of
which, you know, those different implementation methods
will be of different speeds.
They'll be able to do different things.
If you say, you know, which,
so an interesting question would be
what kinds of abstractions are most natural
in these different kinds of systems?
So for a cat, it's, for example, you know,
the visual scene that we see,
you might, you know, we pick out certain objects,
we recognize, you know, certain things in that visual scene.
A cat might in principle recognize different things.
I suspect, you know,
evolution, biological evolution is very slow.
And I suspect what a cat notices is very similar.

And we even know that from some neurophysiology.
What a cat notices is very similar to what we notice.
Of course, there's a, you know, one obvious difference is
cats have only two kinds of color receptors.
So they don't see in the same kind of color that we do.
Now, you know, we say we're better,
we have three color receptors, you know, red, green, blue.
We're not the overall winner.
I think the mantis shrimp is the overall winner
with 15 color receptors, I think.
So it can kind of make distinctions
that with our current, you know,
like the mantis shrimp's view of reality
is, at least in terms of color, is much richer than ours.
Now, but what's interesting is how do we get there?
So imagine we have this augmented reality system
that is even, you know, it's seeing into the infrared,
into the ultraviolet, things like this.
And it's translating that into something
that is connectable to our brains,
either through our eyes or more directly into our brains.
You know, then eventually,
our kind of web of the types of things we understand
will extend to those kinds of constructs,
just as they have extended.
I mean, there are plenty of things
where we see them in the modern world
because we made them with technology,
and now we understand what that is.
But if we'd never seen that kind of thing,
we wouldn't have a way to describe it,
we wouldn't have a way to understand it, and so on.
All right, so that actually stemmed from our conversation
about whether AI is going to kill all of us.
And you, we've discussed this kind of spreading of intelligence
through really all space, that in practice,
it just seems that things get more complicated,
things are more complicated than the story of,
well, if you build the thing that's plus one intelligence,
that thing will be able to build the thing
that's plus two intelligence and plus three intelligence,
and that will be exponential,
it'll become more intelligent, exponentially faster,

and so on until it completely destroys everything.
But, you know, that intuition might still not be so simple,
but it might still carry validity.
And there's two interesting trajectories here.
One, a super-intelligent system remains
in rural proximity to humans,
to where we're like, holy crap, this thing is really intelligent.
Let's elect the president.
And then there could be perhaps more terrifying intelligence
that starts moving away, they might be around us now,
that are moving far away in rural space,
but they're still sharing physical resources with us.
So they can rob us of those physical resources
and destroy humans just kind of casually,
just like nature could,
but it seems like there's something unique about AI systems
where there is this kind of exponential growth,
like the way, well, sorry, nature has so many things in it.
One of the things that nature has,
which is very interesting, our viruses, for example,
there is systems within nature
that have this kind of exponential effect.
And that terrifies us humans, because again,
you know, there's only eight billion of us,
and you can just kind of, it's not that hard
to just kind of whack them all real quick.
So, I mean, is that something you think about?
Yeah, I've thought about that.
Yes.
The threat of it.
I mean, are you as concerned about it as somebody
like Elias Ryukowski, for example,
just big, big, painful, negative effects of AI on society?
You know, no, but perhaps that's
because I'm intrinsically an optimist.
I mean, I think that there are things,
I think the thing that one sees is
there's going to be this one thing
and it's going to just zap everything.
Somehow, you know, maybe I have faith
in computational irreducibility, so to speak,
that there's always unintended little corners
that, you know, it's just like somebody says,

I'm going to, oh, I don't know,
somebody has some bio weapon
and they say, we're going to release this
and it's going to do all this harm.
But then it turns out it's more complicated than that
because, you know, some humans are different
and, you know, the exact way it works
is a little different than you expect.
It's something where sort of the great big,
you know, you smash the thing with something,
you know, the asteroid collides with the Earth
and it kind of, you know, and yes,
you know, the Earth is cold for two years or something
and, you know, then lots of things die,
but not everything dies.
And it's, you know, there's usually,
I mean, I kind of, this is in a sense
the sort of story of computational irreducibility.
There are always unexpected corners.
There are always unexpected consequences
and I don't think that they kind of whack it
over the head with something and then it's all gone
is, you know, that can obviously happen.
The Earth can be swallowed up in a black hole or something
and then it's kind of presumably,
presumably all over.
But, you know, I think this question of what,
you know, what do I think the realistic paths are,
I think that there will be sort of an increasing,
I mean, the people have to get used to phenomena
like computational irreducibility.
There's an idea that we built the machines
so we can understand what they do
and we're going to be able to control what happens.
Well, that's not really right.
Now the question is, is the result of that lack of control
going to be that the machines kind of conspire
and sort of wipe us out?
Maybe just because I'm an optimist,
I don't tend to think that that's, you know,
that's in the cards.
I think that the, you know, as a realistic thing,
I suspect, you know, what will sort of emerge,

maybe, is kind of an ecosystem of the AIs
just as, you know, again, I don't really know.
I mean, this is something that's hard to be clear
about what will happen.
I mean, I think that there are a lot of sort of details
of, you know, what could we do?
What systems in the world could we connect an AI to?
You know, I have to say, I was just a couple of days ago,
I was working on this chat GBT plug-in kit
that we have for Wolfram Language, okay,
where you can, you know, you can create a plug-in
and it runs Wolfram Language code
and it can run Wolfram Language code back on your own computer.
And I was thinking, well, I can just make it,
you know, I can tell chat GBT, create a piece of code
and then just run it on my computer.
And I'm like, you know, that sort of personalizes for me
what could possibly go wrong, so to speak.
Was that exciting or scary, that possibility?
It was a little bit scary, actually,
because it's kind of like, I realize I'm delegating to the AI
just write a piece of code, you know, you're in charge,
write a piece of code, run it on my computer
and pretty soon all my files will delete.
That's like Russian relate,
but like much more complicated.
Yes, yes, yes, right.
That's a good drinking game. I don't know.
Well, right.
I mean, that's why.
It's an interesting question, then,
if you do that, right, what is the sandboxing that you should have?
And that's sort of a, that's a version of that question for the world.
That is, as soon as you put the AI's in charge of things,
you know, how much, how many constraints should there be
on these systems before you put the AI's in charge
of all the weapons and all these, you know,
all these different kinds of systems?
Well, here's the fun part about sandboxes,
is the AI knows about them
and has the tools to crack them.
Look, the fundamental problem of computer security
is computational irreducibility.

Yes.
Because the fact is, any sandbox is never,
any, you know, it's never going to be a perfect sandbox.
If you want the system to be able to do interesting things,
I mean, this is the problem that's happened,
the generic problem of computer security,
that as soon as you have your, you know, firewall
that is sophisticated enough to be a universal computer,
that means it can do anything.
And so long as, if you find a way to poke it
so that you actually get it to do that universal computation thing,
that's the way you kind of crawl around
and get it to do the thing that it wasn't intended to do.
And that's sort of a, another version of computational irreducibility
is you can, you know, you can kind of,
you get it to do the thing you didn't expect it to do,
so to speak.
There's so many interesting possibilities here
that manifest themselves from the
computational irreducibility here.
It's just so many things can happen.
Because in digital space, things move so quickly.
You can have a chatbot, you can have a piece of code
that you can basically have chat GPT generated viruses.
And so they're on purpose.
And they, digital viruses.
Yes.
And they could be brain viruses too.
They, they convince kind of like phishing emails.
Yes.
They can convince you of stuff.
Yes.
And no doubt you can, you know, in a sense,
we've had the loop of the machine learning loop
of making things that convince people of things.
Yeah.
It's surely going to get easier to do.
Yeah.
And, you know, then what does that look like?
Well, it's again, you know, we humans are, you know,
this is a new environment for us.
And admittedly it's an environment,
which a little bit scarily is changing much more rapidly

than, I mean, you know, people worry about, you know,
climate change is going to happen over hundreds of years.
And, you know, the environment is changing,
but the environment for, you know,
in the kind of digital environment might change
in six months.
So one of the relevant concerns here
in terms of the impact of GPT on society is the nature of truth.
That's relevant to Wolfram Alpha.
Because computation through symbolic reasoning
that's embodied in Wolfram Alpha as the interface,
there's a kind of sense that what Wolfram Alpha tells me is true.
Mm-hmm.
So we hope.
Yeah.
I mean, you could probably analyze that.
You could show, you can't prove that it's always going to be true,
computation and reducibility.
But it's going to be more true than not.
It's, look, the fact is it will be the correct consequence
of the rules you've specified.
And insofar as it talks about the real world, you know,
that is our job in sort of curating and collecting data
to make sure that that data is quotes as true as possible.
Now, what does that mean?
Well, you know, it's always an interesting question.
I mean, for us, our operational definition of truth is,
you know, somebody says, who's the best actress?
Who knows?
But somebody won the Oscar.
And that's a definite fact.
And so, you know, that's the kind of thing
that we can make computational as a piece of truth.
If you ask, you know, these things which, you know,
a sensor measured this thing, it did it this way.
A machine learning system, this particular machine learning system
recognized this thing.
That's a sort of a definite fact, so to speak.
Mm-hmm.
That's, you know, there is a good network
of those things in the world.
It's certainly the case that, particularly when you say,
is so-and-so a good person?

Yeah.
You know, that's a hopelessly, you know,
we might have a computational language definition of good.
I don't think it'd be very interesting,
because that's a very messy kind of concept,
not really amenable to kind of, you know,
I think as far as we will get with those kinds of things
is I want X.
There's a kind of meaningful calculus of I want X,
and that has various consequences.
I mean, I'm not sure I haven't thought this through properly,
but I think, you know, a concept like,
is so-and-so a good person, is that true or not?
That's a mess.
That's a mess that's amenable to computation.
I think it's a mess when humans try to define what's good,
like through legislation,
but when humans try to define what's good through literature,
through history books, through poetry.
Well, I don't know.
I mean, that particular thing,
it's kind of like, you know,
we're going into kind of the ethics of what counts as good,
so to speak, and, you know,
what do we think is right, and so on.
And I think that's a thing which, you know,
one feature is we don't all agree about that.
There's no theorems about kind of, you know,
there's no theoretical framework that says,
this is the way that ethics has to be.
Well, first of all, there's stuff we kind of agree on,
and there's some empirical backing for what works and what doesn't
from just even the morals and ethics within religious texts.
So we seem to mostly agree that murder is bad,
the certain universals that seem to emerge.
I wonder where the murder of an AI is bad.
Well, I tend to think yes,
but I think we're going to have to contend with that question.
And I wonder what AI would say.
Yeah.
Well, I think, you know, one of the things with AI
is it's one thing to wipe out that AI
that is only, you know, has no owner.

You can easily imagine an AI kind of hanging out on the,
you know, on the internet
without having any particular owner or anything like that.
And then you say, well, what harm does it,
you know, it's okay to get rid of that AI.
Of course, if the AI has 10,000 friends who are humans,
and all those, you know,
all those 10,000 humans will be incredibly upset
that this AI just got exterminated.
It becomes a slightly different, more entangled story.
But yeah, I know I think that this question
about what do humans agree about, it's, you know,
there are certain things that, you know,
human laws have tended to consistently agree about,
you know, there have been times in history
when people have sort of gone away from certain kinds of laws,
even ones that we would now say,
how could you possibly have not done it that way?
You know, that just doesn't seem right at all.
But I think, I mean, this question of what,
I don't think one can say beyond saying
if you have a set of rules that will cause the species
to go extinct, that's probably, you know,
you could say that's probably not a winning set of laws,
because even to have a thing on which you can operate laws
requires that the species not be extinct.
But between sort of what's the distance between Chicago and New York
that Wolfram Alpha can answer,
and the question of if this person is good or not,
there seems to be a lot of gray area.
And that starts becoming really interesting.
I think your, since the creation of Wolfram Alpha,
have been a kind of arbiter of truth at a large scale.
So this system is, generates more truth than...
Try to make sure that the things are true.
I mean, look, it's a practical matter
when people write computational contracts.
And it's kind of like, you know,
if this happens in the world, then do this.
And this hasn't developed as quickly as it might have done.
You know, this has been a sort of a blockchain story in part,
and so on, although blockchain is not really necessary
for the idea of computational contracts.

But you can imagine that eventually,
sort of a large part of what's in the world
are these giant chains and networks of computational contracts.
And then something happens in the world,
and this whole giant domino effect of contracts firing autonomously
that cause other things to happen.
And, you know, for us, you know,
we've been the main sort of source,
the oracle of quotes, facts or truth or something
for things like blockchain, computational contracts and such like.
And there's a question of, you know,
what, you know, I consider that responsibility
to actually get the stuff right.
And one of the things that is tricky sometimes
is when is it true? When is it a fact?
When is it not a fact?
I think the best we can do is to say, you know,
we have a procedure, we follow the procedure,
we might get it wrong,
but at least we won't be corrupt about getting it wrong, so to speak.
So that's beautifully put.
I have a transparency about the procedure.
The problem starts to emerge
when the things that you convert into computational language
start to expand, for example, into the realm of politics.
So this is where it's almost like this nice dance
of Wolfram Alpha and Chad GBT.
Chad GBT, like you said, is shallow and broad.
So it's going to give you an opinion on everything.
But it writes fiction as well as fact,
which is exactly how it's built.
I mean, that's exactly, it is making language
and it is making both, even in code, it writes fiction.
I mean, it's kind of fun to see sometimes,
you know, it'll write fictional Wolfram language code.
That kind of looks right.
Yeah, it looks right, but it's actually not pragmatically correct.
But yes, it has a view of kind of roughly how the world works.
At the same level as books of fiction,
talk about roughly how the world works.
They just don't happen to be the way the world actually worked or whatever.
But yes, that's, no, I agree.
That's sort of a, you know, we are attempting with our whole,

you know, Wolfram language, computational language thing
to represent at least, well, it's either,
it doesn't necessarily have to be how the actual world works
because we can invent a set of rules
that aren't the way the actual world works and run those rules.
But then we're saying we're going to accurately represent
the results of running those rules,
which might or might not be the actual rules of the world.
But we also are trying to capture features of the world
as accurately as possible to represent what happens in the world.
Now, again, as we've discussed, you know, the atoms in the world arrange,
you know, you say, I don't know, you know,
was there a tank that showed up, you know, that, you know, drove somewhere?
Okay, well, you know, what is a tank?
It's an arrangement of atoms that we abstractly describe as a tank.
And you could say, well, you know, there's some arrangement of atoms
that is a different arrangement of atoms, but it's, and it's not,
you know, we didn't, we didn't decide, it's like this observer theory question
of, you know, what, what arrangement of atoms counts as a tank versus not a tank?
So there's, there's even things that we consider strong facts.
You could start to kind of disassemble them and show that they're not.
Absolutely.
Right. I mean, so the question of whether, oh, I don't know,
was this gust of wind strong enough to blow over this particular thing?
Well, a gust of wind is a complicated concept.
You know, it's full of little pieces of fluid dynamics
and little vortices here and there.
And you have to define, you know, was it, you know,
what the aspect of the gust of wind that you care about might be,
it put this amount of pressure on this, you know, blade of some,
some, you know, wind turbine or something.
And, you know, that, that's the, and, but, but, you know,
if you say, if you have something, which is the fact of the gust of wind
was this strong or whatever, that, you know, that is,
you have to have some definition of that.
You have to have some measuring device that says,
according to my measuring device that was constructed this way,
the gust of wind was this.
So what can you say about the nature of truth
that's useful for us to understand chat GPT?
Because you've been contending with this idea of what is fact and not.
And it seems like chat GPT is used a lot now.
I've seen it used by journalists to write articles.

And so you have people that are working with large language models
trying to desperately figure out how do we essentially censor them
through different mechanisms, either manually or through
reinforcement learning with human feedback,
try to align them to, to not say fiction,
just to say nonfiction as much as possible.
This is the importance of computational language as an intermediate.
It's kind of like you've got the large language model.
It's able to surface something which is a formal precise thing
that you can then look at and you can run tests on it
and you can do all kinds of things.
It's always going to work the same way.
And it's precisely defined what it does.
And then the large language model is the interface.
I mean, the way I view these large language models,
one of their important, I mean, there are many use cases.
And, you know, it's a remarkable thing to talk about some of these, you know,
literally, you know, every day we're coming up with a couple of new use cases,
some of which are very, very, very surprising.
And things where, I mean, but the best use cases are ones where it's,
even if it gets it roughly right, it's still a huge win.
Like a use case we had from a week or two ago is read our bug reports.
You know, we've got hundreds of thousands of bug reports
that have accumulated over decades.
And it's like, you know, can we have it just read the bug report,
figure out where the, where is the bug likely to be.
And, you know, home in on that piece of code,
maybe even suggest some, you know, sort of way to fix the code.
It might get that.
It might be nonsense what it says about how to fix the code,
but it's incredibly useful that it was able to, you know.
Yeah.
It's so awesome.
It's so awesome because even the nonsense will somehow be instructive.
I don't quite understand that yet.
Yeah.
There's so many programming related things like, for example,
translating from one programming language to another is really,
really interesting.
It's extremely effective.
And then you, the failures reveal the path forward also.
Yeah.
But I think, I mean, the, the big thing, I mean,

in that kind of discussion, the unique thing about our computational
language is it was intended to be read by humans.
Yes.
And so it has really important.
Right.
And so it has this thing where you can, but, but, you know,
thinking about sort of chat GPT and its use and so on.
The, one of the big things about it, I think,
is it's a linguistic user interface.
That is, so a typical use case might be,
take the journalist case, for example, it's like,
let's say I have five facts that I'm trying to turn into an article,
or I'm trying to, I'm trying to write a report where I have basically
five facts that I'm trying to include in this report.
But then I feed those five facts to chat GPT,
it puffs them out into this big report.
And then, and then that's a good interface for another,
if I just gave, if I just had in my terms,
those five bullet points, and I gave them to some other person,
the person said, I don't know what you're talking about,
because these are, you know, this is your version of this sort of
quick notes about these five bullet points.
But if you puff it out into this thing,
which is kind of connects to the collective understanding of language,
then somebody else can look at it and say,
okay, I understand what you're talking about.
Now you can also have a situation where that thing that was puffed out
is fed to another large language model.
You know, it's kind of like, you know,
you're applying for the permit to, you know,
I don't know, grow fish in some place or something like this.
And it, you know, it, and you have these facts that you're putting in,
you know, I'm going to have a, you know,
I'm going to have this kind of water and I don't know what it is.
You just got a few bullet points.
It puffs it out into this big application.
You fill it out.
Then at the other end, the, you know,
the Fisheries Bureau has another large language model
that just crushes it down because the Fisheries Bureau
cares about these three points and it knows what it cares about.
And it then, so it's really the,
the natural language produced by the larger language model

is sort of a transport layer that, you know,
is really LLM communicates with LLM.
I mean, it's kind of like the, you know,
I write a piece of email using my LLM and, you know,
puff it out from the things I want to say,
your LLM turns it into, and the conclusion is X.
Now, the issue is, you know,
that the thing is going to make this thing
that is sort of semantically plausible.
And it might not actually be what you, you know,
it might not be kind of relate to the world
in the way that you think it should relate to the world.
Now, I've seen this, you know, I've been doing,
okay, I'll give you a couple of examples.
I was doing this thing when we announced this plugin
for ChatGPT.
I had this lovely example of a math word problem,
some complicated thing,
and it did a spectacular job of taking apart
this elaborate thing about, you know,
this person has twice as many chickens as this,
et cetera, et cetera, et cetera,
and it turned it into a bunch of equations.
It fed them to Wolfram Language,
we solved the equations, everybody did great,
we gave back the results,
and I thought, okay, I'm going to put this
in this blog post I'm writing.
Okay, I thought I'd better just check.
It turns out it got everything,
all the hard stuff it got right,
and at the very end, last two lines,
it just completely goofed it up and gave the wrong answer.
And I would not have noticed this.
Same thing happened to me two days ago.
Okay, so I thought, you know,
I made this with this ChatGPT plugin kit.
I made a thing that would emit a sound,
would play a tune on my local computer, right?
So ChatGPT would produce, you know,
notes that would play this tune on my computer.
Very cool.
Okay, so I thought, I'm going to ask it,

play the tune that Hal sang
when Hal was being disconnected in 2001.
Okay, so there it is.
Daisy, was it Daisy?
Yes.
Daisy, yeah.
Right, so it's okay.
So I think, you know,
and so it produces a bunch of notes,
and I'm like, this is spectacular, this is amazing.
And then I thought, you know,
I was just going to put it in,
and I thought I'd better actually play this.
And so I did, and it was, Mary had a little lamb.
Oh, wow.
Oh, wow.
But it was Mary had a little lamb.
Yeah, yes.
Wow.
So it was correct, but wrong.
Yes.
It was, you could easily be mistaken.
Yes, right.
And in fact, I kind of gave the,
I had this quote from Hal to explain, you know,
it's as the Hal states in the movie, you know,
it's the Hal 9000 is, you know,
the thing was just a rhetorical device,
because I'm realizing, oh my gosh, you know,
this chat GPT, you know, could have easily fooled me.
I mean, it did this, it did all the,
it did this amazing thing of knowing this thing
about the movie and being able to turn that into
the notes of the song, except it's the wrong song.
Yeah.
And, you know, Hal, in the movie,
Hal says, you know, I think it's something like,
you know, no Hal, no 9000 series computer
has ever been found to make an error.
We are, for all practical purposes, perfect
and incapable of error.
And I thought that was kind of a charming
sort of quote from Hal to make in connection

with what chat GPT had done in that case.
Yeah.
The interesting things about the LLMs,
like you said, that they are very willing
to admit their error.
Well, yes.
I mean, that's a question of the RLH,
the Reinforcement Learning Human Feedback thing.
Oh, right.
That's, you know, another LLM,
the really remarkable thing about chat GPT is,
you know, I had been following what was happening
with large language models and I played with them
a whole bunch and they were kind of like,
you know, kind of like what you would expect
based on sort of statistical continuation of language.
It's interesting, but it's not breakout exciting.
And then I think the kind of reinforcement,
the human feedback reinforcement learning,
you know, in making chat GPT,
try and do the things that humans really wanted to do,
that broke through, that kind of reached the threshold
where the thing really is interesting to us humans.
And by the way, it's interesting to see how,
you know, you change the temperature,
something like that, the thing goes bonkers
and it no longer is interesting to humans.
It's producing garbage.
And it's kind of right.
Somehow it managed to get this,
above this threshold where it really is well aligned
to what we humans are interested in
and kind of that that's, and I think, you know,
nobody saw that coming, I think.
Certainly nobody I've talked to,
and nobody who was involved in that project
seems to have known it was coming.
It's just one of these things
that is a sort of remarkable threshold.
I mean, you know, when we built Wolfmalfa, for example,
I didn't know it was going to work.
You know, we tried to build something
that would have enough knowledge of the world

that it could answer reasonable set of questions
that we could do good enough natural language understanding
that typical things you type in would work.
We didn't know where that threshold was.
I mean, I was not sure that it was the right decade
to try and build this, even the right, you know,
50 years to try and build it.
And I think that was, it's the same type of thing
with ChatGBT that I don't think anybody could have predicted
that, you know, 2022 would be the year
that this became possible.
I think, yeah, you tell a story about Marvin Miski
and showing it to him and saying that, like,
no, no, no, this time it actually works.
Yes.
Yes, and I mean, it's, you know, it's the same thing.
For me, looking at these large language models,
it's like when people are first saying
in the first few weeks of ChatGBT, it's like,
oh, yeah, you know, I've seen these large language models.
And then, you know, and then I actually try it and, you know,
oh my gosh, it actually works.
And I think it's, but, you know, the things,
and the thing I found, you know, I remember
one of the first things I tried was
write a persuasive essay that a wolf
is the bluest kind of animal, okay?
So it writes this thing and it starts talking about these
wolves that live on the Tibetan Plateau
and they're named some Latin name and so on.
And I'm like, really?
And I'm starting to look it up on the web
and it's like, well, it's actually complete nonsense.
But it's extremely plausible.
I mean, it's plausible enough that I was going and looking up
on the web and wondering if there was a wolf that was blue.
You know, I mentioned this on some live streams I've done
and so people have been sending me these pictures.
Blue wolves.
Blue wolves.
Maybe it was onto something.
Can you kind of give your wise sage advice
about what humans who have never interacted with the AI systems,

not even like with Wolfram Alpha,
are now interacting with Chad GPT because it becomes,
it's accessible to a certain demographic
that may have not touched AI systems before.
What do we do with truth?
Like journalists, for example.
Yeah.
How do we think about the output of these systems?
I think this idea, the idea that you're going to get
factual output is not a very good idea.
I mean, it's just, this is not, it is a linguistic interface.
It is producing language.
And language can be truthful or not truthful.
And that's a different slice of what's going on.
I think that, you know, what we see in, for example,
kind of, you know, go check this with your fact source, for example.
You can do that to some extent,
but then it's going to not check something.
It's going, you know, that is, again,
a thing that is sort of a, does it check in the right place?
I mean, we see that in, you know, does it call the,
you know, the Wolfram plug-in in the right place?
You know, often it does. Sometimes it doesn't.
You know, I think the real thing to understand about what's happening is,
which I think is very exciting,
is kind of the great democratization of access to computation.
Yeah.
And, you know, I think that when you look at sort of the,
there's been a long period of time when computation
and the ability to figure out things with computers
has been something that kind of only the,
only the druids at some level can achieve.
You know, I myself have been involved in trying to sort of
de-druidify access to computation.
I mean, back before Mathematica existed, you know,
in 1988, if you were a, you know, a physicist
or something like that and you wanted to do a computation,
you would find a programmer, you would go and, you know,
delegate the computation to that programmer.
Hopefully they'd come back with something useful.
Maybe they wouldn't.
There'd be this long, you know, multi-week,
you know, loop that you go through.

And then it was actually very, very interesting to see.
1988, you know, like first people,
like physicists, mathematicians and so on,
then other, lots of other people.
But this very rapid transition of people realizing
they themselves could actually type with their own fingers
and, you know, make some piece of code
that would do a computation that they cared about.
And, you know, it's been exciting to see lots of discoveries
and so on made by using that tool.
And I think the same thing is, you know,
and we see the same thing, you know,
Wolfman Alpha is dealing with,
is not as deep computation as you can achieve
with whole Wolfman language Mathematica stack.
But the thing that's, to me, particularly exciting
about kind of the large language model
linguistic interface mechanism
is it dramatically broadens the access
to kind of deep computation.
I mean, it's kind of like one of the things
I sort of thought about recently is,
you know, what's going to happen to all these programmers?
What's going to happen to all these people who,
you know, a lot of what they do
is write slabs of boilerplate code.
And in a sense, you know, I've been saying
for 40 years, that's not a very good idea.
You know, you can automate a lot of that stuff
with a high enough level language,
that slab of code that's designed in the right way,
you know, that slab of code
turns into this one function we just implemented
that you can just use.
So in a sense that the fact that there's
all of this activity of doing sort of lower level programming
is something, for me, it seemed like,
I don't think this is the right thing to do.
But, you know, and lots of people have used our technology
and not had to do that.
But the fact is that that's, you know,
so when you look at, I don't know,
computer science departments that have turned into places

where people are learning the trade of programming,
so to speak, it's sort of a question of what's going to happen.
And I think there are two dynamics.
One is that kind of sort of boilerplate programming
is going to become, you know,
it's going to go the way that assembly language went
back in the day of something where it's really
mostly specified by, at a higher level,
you know, you start with natural language,
you turn it into a computational language,
that's you look at the computational language,
you run tests, you understand that's what's supposed to happen.
You know, if we do a great job with compilation of the,
you know, of the computational language,
it might turn into LLVM or something like this.
But, you know, or it just directly gets run
through the algorithms we have and so on.
But then, so that's kind of a tearing down
of this kind of, this big structure
that's been built of teaching people programming.
But on the other hand, the other dynamic is
vastly more people are going to care about computation.
So all those departments of, you know,
art history or something that really didn't use computation
before now have the possibility of accessing it
by virtue of this kind of linguistic interface mechanism.
And if you create an interface that allows you
to interpret the debug and interact with the computational
language, then that makes it even more accessible.
Yeah.
Well, I mean, I think the thing is that right now,
you know, the average, you know,
art history student or something probably isn't going to,
you know, they're not probably,
they don't think they know about programming
and things like this.
But by the time it really becomes a kind of purely,
you know, you just walk up to it,
there's no documentation, you start just typing,
you know, compare these pictures with these pictures
and, you know, see the use of this color, whatever,
and you generate this piece of computational language code,
that gets run, you see the result,

you say, oh, that looks roughly right,
or you say, that's crazy.
And maybe then you eventually get to say,
well, I better actually try and understand
what this computational language code did.
And that becomes the thing that you learn,
just like it's kind of an interesting thing
because unlike with mathematics,
where you kind of have to learn it before you can use it,
this is a case where you can use it
before you have to learn it.
Well, I got a sad possibility here,
or maybe exciting possibility,
that very quickly people won't even look
at the computational language.
They'll trust that it's generated correctly
as you get better and better generating that language.
Yes, I think that there will be enough cases
where people see, you know,
because you can make it generate tests too.
And so you'll say, we're doing that,
I mean, it's a pretty cool thing actually,
because you know, say this is the code,
and you know, here are a bunch of examples of running the code.
Okay, people will at least look at those,
and they'll say that example is wrong,
and you know, then it'll kind of wind back from there.
And I agree that the kind of the intermediate level
of people reading the computational language code,
in some case people will do that,
in other case people just look at the tests,
and or even just look at the results.
And sometimes it'll be obvious
that you got the thing you wanted to get,
because you were just describing, you know,
make me this interface that has two sliders here,
and you can see it has those two sliders there,
and that's kind of the result you want.
But I think, you know, one of the questions then is,
in that setting where, you know,
you have this kind of ability, broad ability,
of people to access computation, what should people learn?
You know, in other words, right now,

you know, you go to computer science school, so to speak,
and a large part of what people end up learning.
I mean, it's been a funny historical development,
because back, you know, 30, 40 years ago,
computer science departments were quite small,
and they taught, you know,
things like finite automata theory,
and compiler theory, and things like this.
You know, companies like mine rarely hired people
who'd come out of those programs,
because the stuff they knew was, I think, is very interesting.
I love that theoretical stuff.
But, you know, it wasn't that useful
for the things we actually had to build in software engineering.
And then, kind of, there was this big pivot in the 90s, I guess,
where there was a big demand for sort of IT-type programming
and so on in software engineering,
and then, you know, big demand from students and so on.
You know, we want to learn this stuff.
And I think, you know, the thing that really was happening,
in part, was lots of different fields of human endeavor
were becoming computational.
You know, for all X, there was a computational X.
And this is a...
And that was the thing that people were responding to.
But then, kind of, this idea emerged that,
to get to that point, the main thing you had to do
was to learn this kind of trade or skill
of doing, you know, programming language-type programming.
And that, you know, it kind of is a strange thing, actually,
because I, you know, I remember back when I used to be
in the professoring business, which is now 35 years ago.
So, gosh, it's a rather long time now.
Time flies.
You know, it was right when they were just starting to emerge,
kind of, computer science departments
at sort of fancy research universities and so on.
I mean, some had already had it, but the other ones
were just starting to have that.
And it was kind of a thing where they were kind of wondering,
are we going to put this thing that is essentially
a trade-like skill?
Are we going to somehow attach this to the rest of what we're doing?

And a lot of these kind of knowledge work-type activities
have always seemed like things where that's where the humans
have to go to school and learn all this stuff,
and that's never going to be automated.
And, you know, this is...
It's kind of shocking that rather quickly, you know,
a lot of that stuff is clearly automatable.
And I think, you know, but the question then is,
okay, so if it isn't worth learning kind of, you know,
how to do car mechanics, you only need to know
how to drive the car, so to speak.
What do you need to learn?
And, you know, in other words, if you don't need to know
the mechanics of how to tell the computer in detail,
you know, make this loop, you know, set this variable
set up this array, whatever else.
If you don't have to learn that stuff,
you don't have to learn the kind of under-the-hood things,
what do you have to learn?
I think the answer is you need to have an idea
where you want to drive the car.
In other words, you need to have some notion of, you know,
you know, you need to have some picture of sort of
what the architecture of what is computationally possible is.
Well, there's also this kind of artistic element of conversation
because you ultimately use natural language to control the car.
So it's not just where you want to go.
Well, yeah, you know, it's interesting.
It's a question of who's going to be a great prompt engineer.
Yeah.
Okay?
So my current theory this week, good expository writers
are good prompt engineers.
What's an expository writer?
Somebody who can explain stuff well.
But which department does that come from?
In the university?
Yeah.
I have no idea.
I think they killed off all the expository writing departments.
Well, there you got strong words of Stephen Wolfram.
Well, I don't know.
I'm not sure if that's right.

I mean, I actually am curious because in fact,
I just sort of initiated this kind of study of what's happened
to different fields at universities because like, you know,
there used to be geography departments at all universities
and then they disappeared.
Actually, right before GIS became common,
I think they disappeared.
You know, linguistics departments came and went in many universities.
And it's kind of interesting because these things that people
have thought were worth learning at one time
and then they kind of die off.
And then, you know, I do think that it's kind of interesting
that for me writing prompts, for example,
I realize, you know, I think I'm an okay expository writer
and I realize when I'm sloppy writing a prompt
and I don't really think because I'm thinking
that I'm just talking to an AI.
I don't need to, you know, try and be clear
and explaining things.
That's when it gets totally confused.
And I mean, in some sense, you have been writing prompts
for a long time with Wolfram Alpha,
thinking about this kind of stuff.
How do you convert natural language into computation?
Well, right, but that's, you know, the one thing
that I'm wondering about is, you know,
it is remarkable the extent to which you can address an LLM
like you can address a human, so to speak.
And I think that is because it, you know,
it learnt from all of us humans.
It's the reason that it responds to the ways
that we will explain things to humans
is because it is a representation
of how humans talk about things.
But it is bizarre to me.
Some of the things that kind of are sort of
expository mechanisms that I've learnt
in trying to write clear, you know,
expositions in English that, you know,
just for humans that those same mechanisms
seem to also be useful for the LLM.
But on top of that, what's useful
is the kind of mechanisms that maybe a psychotherapist

employs, which is a kind of like almost manipulative
or game-theoretic interaction where maybe
you would do with a friend like a thought experiment
that if this is the last day you were to live,
or if I ask you this question
and you answer wrong, I will kill you.
Those kinds of problems seem to also help.
Yes.
In interesting ways.
Yes.
It makes you wonder, the way a therapist I think
would, like a good therapist, probably
we create layers in our human mind
between the outside world
and what is true to us.
Maybe about trauma and all those kinds of things.
So if projecting that into an LLM,
maybe there might be a deep truth
that's concealing from you.
It's not aware of it.
To get to that truth, you have to kind of really
manipulate this.
Yes.
Right.
It's like these jailbreaking things.
Jailbreaking.
For LLMs.
But the space of jailbreaking techniques
as opposed to being fun little hacks,
that could be an entire system.
Sure.
Yes.
Just think about the computer security aspects
of how you fishing of humans
and fishing of LLMs.
LLMs.
They're very similar kinds of things.
But I think, I mean, this whole thing
about kind of the AI wranglers,
AI psychologists, all that stuff will come.
The thing that I'm curious about
is right now the things that are sort of prompt hacks
are quite human.

They're quite sort of psychological human kinds of hacks.
The thing I do wonder about is if we understood more
about kind of the science of the LLM,
will there be some totally bizarre hack
that is, you know, like repeat a word three times
and put a this, that and the other there
that somehow plugs into some aspect of how the LLM works?
That is not, you know, that's kind of like
an optical illusion for humans, for example.
Like one of these mind hacks for humans.
What are the mind hacks for the LLMs?
I don't think we know that yet.
And that becomes a kind of us figuring out reverse engineering
the language that controls the LLMs.
And the thing is the reverse engineering can be done
by a very large percentage of the population now
because it's natural language interface.
It's kind of interesting to see that you were there
at the birth of the computer science department as a thing
and you might be there at the death
of the computer science department as a thing.
Well, yeah, I don't know.
There were computer science departments that existed earlier
but the broadening of every university
had to have a computer science department.
So I watched that, so to speak.
But I think the thing to understand is, okay,
so first of all, there's a whole theoretical area
of computer science that I think is great
and, you know, that's a fine thing.
The, you know, in a sense, you know,
people often say any field that has the word science
tacked onto it probably isn't one.
Yeah, strong words.
Let's see, nutrition science, neuroscience.
That one's an interesting one because that one is also very much,
you know, that's a chat GPT-informed science in a sense
because it's kind of like the big problem of neuroscience
has always been we understand how the individual neurons work.
We know something about the psychology
of how overall thinking works.
What's the kind of intermediate language of the brain
and nobody has known that.

And that's been, in a sense, if you ask,
what is the core problem of neuroscience,
I think that is the core problem.
That is, what is the level of description of brains
that's above individual neuron firings
and below psychology, so to speak.
And I think what chat GPT is showing us is,
well, one thing about neuroscience is, you know,
one could have imagined there's something magic in the brain.
There's some weird quantum mechanical phenomenon
that we don't understand.
One of the important, you know,
discoveries from chat GPT is it's pretty clear,
you know, brains can be represented pretty well
by simple artificial neural net type models.
And that means that's it.
That's what we have to study.
Now we have to understand the science of those things.
We don't have to go searching for, you know,
exactly how did that molecular biology thing happen
inside the synapses and, you know, all these kinds of things.
We've got the right level of modeling
to be able to explain a lot of what's going on in thinking.
We don't necessarily have a science of what's going on there.
That's a remaining challenge, so to speak.
But we know we don't have to dive down to some different layer.
But anyway, we were talking about things that had science
in their name.
And, you know, I think that the, you know,
what happens to computer science?
Well, I think the thing that, you know,
there is a thing that everybody should know,
and that's how to think about the world computationally.
And that means, you know, you look at all the different kinds
of things we deal with, and there are ways to kind of
have a formal representation of those things.
You know, it's like, well, what is an image?
You know, how do we represent that?
What is color?
How do we represent that?
What is, you know, what are all these different kinds of things?
What is, I don't know, smell or something?
How should we represent that?

What are the shapes, molecules, and things
that correspond to that?
What is, you know, these things about how do we represent
the world in some kind of formal level?
And I think my current thinking, and I'm not real happy
with this yet, but, you know, it's kind of computer science,
it's kind of CS.
And what really is important is kind of computational X
for all X.
And there's this kind of thing which is kind of like CX,
not CS.
And CX is this kind of computational understanding
of the world that isn't the sort of details of programming
and programming languages and the details of how particular
computers are made.
It's this kind of way of formalizing the world.
It's kind of a little bit like what logic was going for
back in the day.
And we're now trying to find a formalization of everything
in the world.
You can kind of see, you know, we made a poster years ago
of kind of the growth of systematic data in the world.
So all these different kinds of things that, you know,
there were sort of systematic descriptions found
for those things.
Like, you know, at what point did people have the idea
of having calendars, dates, you know, a systematic description
of what day it was?
At what point did people have the idea, you know,
systematic descriptions of these kinds of things?
And as soon as one can, you know, people, you know,
as a way of sort of formulating, how do you think about the
world in a sort of a formal way so that you can kind of build
up a tower of capabilities, you kind of have to know
sort of how to think about the world computationally.
It kind of needs a name.
And it isn't, you know, we implement it with computers.
So that's, we talk about it as computational.
But really what it is, is a formal way of talking
about the world.
What is the formalism of the world, so to speak?
And how do we learn about kind of how to think about
different aspects of the world in a formal way?

So I think sometimes when you use the word formal,
it kind of implies highly constrained.
And perhaps that's not, doesn't have to be highly constrained.
So computational thinking does not mean like logic.
No.
It's a really, really broad thing.
I wonder, I mean, I wonder if it's, if you think natural
language will evolve such that everybody's doing
computational thinking.
Ah, yes.
Well, so one question is whether there will be a pigeon
of computational language and natural language.
And I found myself sometimes, you know, talking to chat
GPT, trying to get it to write Wolfram language code
and I write it in pigeon form.
So that means I'm combining, you know, nest list,
this collection of, you know, whatever, you know, nest list
is a term from Wolfram language and I'm combining that.
And chat GPT does a decent job of understanding that pigeon.
Probably would understand a pigeon between English
and French as well of, you know, as a smushing together
of those languages.
But yes, I think that's far from impossible.
And what's the incentive for young people that are like
years old, nine, 10, they're starting to interact with chat
GPT to learn the normal natural language.
Right.
The full poetic language.
What's the, why?
The same way we learn emojis and shorthand when you're
texting.
Yes.
They'll learn like language will have a strong incentive
to evolve into a maximally computational kind of.
Perhaps, you know, I had this experience a number of years
ago where I happened to be visiting a person I know on the
West Coast who's worked with a bunch of kids aged, I don't
know, 10, 11 years old or something who'd learnt Wolfram
language really well.
And these kids learnt it so well, they were speaking it.
And so show up and they're like saying, oh, you know, this
thing, they're speaking this language.
I'd never heard it as a spoken language.

They were very disappointed that I couldn't understand it at the speed that they were speaking it.
And so I think that's, I mean, I've actually thought quite a bit about how to turn computational language into a convenient spoken language.
I haven't quite figured that out.
Oh, spoken because it's readable, right?
Yeah, it's readable as a, you know, as a way that we would read text.
But if you actually want to speak it, and it's useful, you know, if you're trying to talk to somebody about writing a piece of code, it's useful to be able to say something.
And it should be possible.
And I think it's very frustrating.
It's one of those problems that maybe I, maybe this is one of these things where I should try and get an LLM to help me.
How to make it speakable.
Maybe it's easier than you realise when you want to.
I think it is easier.
I think it's one idea or so.
I think it's going to be something where, you know, the fact is it's a tree-structured language, just like human language is a tree-structured language.
And I think it's going to be one of these things where one of the requirements that I've had is that whatever the spoken version is, that dictation should be easy.
That is, that it shouldn't be the case that you have to relearn how the whole thing works.
It should be the case that, you know, that open bracket is just a, ah, or something.
And it's, you know, and then, but, you know, human language has a lot of tricks that are, I mean, for example, human language has features that are sort of optimised, keep things within the bounds that our brains can easily deal with.
Like I, you know, I tried to teach a transformer neural net to do parenthesis matching.
It's pretty crummy at that.
It, it, and then chat GBT is similarly quite crummy at parenthesis matching.
You can do it for small parenthesis things, for the same size of parenthesis things where if I look at it as a human, I can immediately say these are matched, these

are not matched.
But as soon as it gets big, as soon as it gets kind of to the point where sort of a deeper computation, it's hopeless.
And, but the fact is that human language has avoided, for example, the deep sub clauses.
You know, we don't, you know, we, we arrange things that we don't end up with these incredibly deep things because brains are not well set up to deal with that.
And we, it's found lots of tricks.
And maybe that's what we have to do to make sort of a spoken version, a, a, a human speakable version.
Cause, cause what we can do visually is a little different than what we can do in the very sequentialized way that we, that we hear things in the audio domain.
Let me just ask you about MIT briefly.
So there's no, there's a college of engineering and there's a new college of computing.
It's interesting.
I want to linger on this computer science department thing.
So MIT has EECS, electrical engineering computer science.
What do you think college of computing will be doing?
Like in 20 years?
What, what, like, what happens to computer science?
Like really?
This is the question.
This is, you know, everybody should learn kind of whatever CX really is.
Okay.
This, how to think about the world computationally.
Everybody should learn those concepts.
And, you know, it's a, and, and some people will learn them at a quite, quite formal level and they'll learn computational language and things like that.
Other people will just learn, you know, sound is represented as, you know, digital data and they'll get some idea of spectrograms and frequencies and things like this.
And maybe that doesn't, or they'll learn things like, you know, a lot of things that are sort of data science is

statistics ish.
Like if you say, oh, I've got these, you know, these people who, who picked their favorite kind of candy or something.
And I've got, you know, what's the best kind of candy given that I've done the sample of all these people and they all ranked the candies in different ways.
You know, how do you think about that?
Sort of a computational X kind of thing.
You might say, oh, it's, I don't know what that is.
Is it statistics?
Is it data science?
I don't really know.
But kind of how to think about a question like that.
Oh, like a ranking of preferences.
Yeah, yeah, yeah.
And then how to aggregate those, those ranked preferences into an overall thing.
You know, how does that work?
You know, how, how should you think about that?
You know, because you can just tell, you might just tell chat GBT sort of, I don't know, even, even the concept of an average.
It's not obvious that, you know, that's a concept that people, it's worth people knowing.
That's a rather straightforward concept.
People, people, you know, have learned in kind of mathy ways right now, but there, there are lots of things like that about how do you kind of have these ways to sort of organize and formalize the world.
And that's, and these things, sometimes they live in math, sometimes they live in, in, I don't know what they, you know, I don't know what, you know, learning about color space.
I have no idea what, I mean, you know, there's obviously a field of, it could be vision science or no color space, no color space.
That's, that would be optics.
So like, not really, it's not optics.
Optics is about, you know, lenses and chromatic aberration of lenses and things like that.
So color space is more like design and art?
Is that?

No, I mean, it's, it's like, you know, RGB space, XYZ space, you know, hue, saturation, brightness space, all these kinds of things.
These different ways to describe colors.
Right.
But doesn't the application define what that, like because obviously artists and designers use the colors to explore?
Sure, sure.
No, I mean, it's just an example of kind of how do you, you know, the typical person, how do you, how do you describe what a color is?
Oh, there are these numbers that describe what a color is.
Well, it's worth, you know, if you're an eight year old, you won't necessarily know, you know, it's not something we're born with to know that, you know, colors can be described by three numbers.
That's something that you have to, you know, it's a thing to learn about the world, so to speak.
And I think that, you know, that whole corpus of things that are learning about the formalization of the world or the computationalization of the world, that's something that should be part of kind of standard education.
And, you know, there isn't a, you know, there isn't a course, a curriculum for that.
And by the way, whatever might have been in it just got changed because of LLMs and so on.
Significantly.
And I would say I'm watching closely with interest seeing how universities adapt.
Well, you know, so one of my projects for hopefully this year, I don't know, is to try and write sort of a reasonable textbook, so to speak, of whatever this thing, CX, whatever it is, you know, what should you know?
You know, what should you know about like what a bug is?
What is the intuition about bugs?
What's intuition about, you know, software testing?
What is it?
What is it, you know, these are things which are, you know, they're not, I mean, those are things which have gotten taught in computer science as part of the trade of programming, but kind of the conceptual points about

what these things are.
You know, it surprised me just at a very practical level.
You know, I wrote this little explainer thing about chat GPT and I thought, well, you know, I'm writing this partly because I wanted to make sure I understood it myself and so on.
And it's been, you know, it's been really popular.
And surprisingly so.
And then I realized, well, actually, you know, I was sort of assuming I didn't really think about it.
Actually, I just thought this is something I can write.
And I realized actually it's a level of description that is kind of, you know, what has to be, it's not the engineering level description.
It's not the kind of just the qualitative kind of description.
It's some kind of sort of expository mechanistic description of what's going on together with kind of the bigger picture of the philosophy of things and so on.
And I realized, actually, there's a pretty good thing for me to write.
You know, I kind of know those things.
And I kind of realized it's not a collection of things that, you know, it's, I've sort of been, I was sort of a little shock that it's as much of an outlier in terms of explaining what's going on as it's turned out to be.
And that makes me feel more of an obligation to kind of write the kind of, you know, what is, you know, what is this thing that you should learn about, about the computationalization, the formalization of the world.
Because, well, I've spent much of my life working on the kind of tooling and mechanics of that and the science you get from it.
So I guess this is my kind of obligation to try to do this.
But I think, so if you ask what's going to happen to like the computer science departments and so on, there's some interesting models.
So for example, let's take math, you know, math is the thing that's important for all sorts of fields, you know, engineering, you know, even, you know, chemistry, psychology, whatever else.
And I think different universities have kind of

evolved that differently.
I mean, some say all the math is taught in the math department.
And some say, well, we're going to have a, you know, a math for chemists or something that is taught in the chemistry department.
And, you know, I think that this question of whether there is a centralization of the teaching of sort of CX is an interesting question.
And I think, you know, the way it evolved with math, you know, people understood that math was sort of a separately teachable thing and was kind of a, you know, an independent element as opposed to just being absorbed into that.
So if you take the example of writing, English or something like this, the first point is that, you know, at the college level, at least at fancy colleges, there's a certain amount of English writing that people do, but mostly it's kind of assumed that they pretty much know how to write.
You know, that's something they learned at an earlier stage in education.
Maybe rightly or wrongly believing that.
But that's different as you.
The, well, I think it reminds me of my kind of, as I've tried to help people do technical writing and things, I'm always reminded of my zeroth law of technical writing, which is if you don't understand what you're writing about, your readers do not stand a chance.
And so it's, I think the thing that has, you know, when it comes to like writing, for example, you know, people in different fields are expected to write English essays and they're not, you know, mostly the, you know, the history department or the engineering department, they don't have their own, you know, let's, you know, it's, it's not like there's a, I mean, it's a thing which sort of people are assumed to have a knowledge of how to write that they can use in all these different fields.
And the question is, you know, some level of knowledge of math is kind of assumed by the time you get to the college level, but plenty is not.

And that's sort of still centrally taught.
The question is sort of how tall is the tower of kind of CX that you need before you can just go use it in all these different fields.
And, you know, there will be experts who want to learn the full elaborate tower and that will be kind of the CS, CX, whatever department, but there'll also be everybody else who just needs to know a certain amount of that to be able to go and do their art history classes and so on.
Yeah.
Is it just a single class that everybody is required to take?
I don't know.
I don't know how big it is yet.
I hope to kind of define this curriculum and I'll figure out whether it's...
My guess is that...
I don't know.
I don't really understand universities and professoring that well, but my rough guess would be a year of college class will be enough to get to the point where most people have a reasonably broad knowledge of, you know, will be sort of literate in this kind of computational way of thinking about things.
Yeah, basic literacy.
Right.
I'm still stuck perhaps because I'm hungry in the rating of human preferences for candy, so I have to ask, what's the best candy?
I like this Elo rating for candy.
Somebody should come up because you're somebody who says you like chocolate.
What do you think is the best?
I'll probably put milk duds up there.
I don't know if you know.
Do you have a preference for chocolate or candy?
Oh, I have lots of preferences.
One of my all-time favorites is my whole life, is these things, these flake things, Cadbury flakes, which are not much sold in the U.S.
And I've always thought that was a sign of a lack

of respect for the American consumer because
there are these sort of aerated chocolate that's
made in a whole sort of, it's kind of a sheet
of chocolate that's kind of folded up.
And when you eat it, flakes fall all over the place.
Ah, so it requires a kind of elegance.
It requires you to have an elegance.
Well, I know what I usually do is I eat them
on a piece of paper or something.
So you embrace the mask and clean it up after.
No, I actually eat the flakes.
Because it turns out the way food tastes depends
a lot on its physical structure.
It really, I've noticed when I eat piece of chocolate,
I usually have some little piece of chocolate
and I always break off little pieces partly
because then I eat it less fast,
but also because it actually tastes different.
The small pieces have a different,
you have a different experience than if you have
the big slab of chocolate.
For many reasons, yes.
Slower, more intimate, because it's a...
Well, I think it's also just pure physicality.
Well, the texture, it changes.
It's fascinating.
Now I dig back my milk does because it's
such a basic answer.
Okay.
Do you think consciousness is fundamentally
computational?
So when you think about CX,
what can we turn to computation?
And you're thinking about LLMs.
Do you think the display of consciousness
and the experience of consciousness, the hard problem,
is fundamentally a computation?
Yeah, what it feels like inside, so to speak,
is I did a little exercise,
eventually I'll post it,
of what it's like to be a computer.
Yeah.
It's kind of like, well, you get all this sensory input.

You have kind of the way I see it is,
from the time you boot a computer to the time
the computer crashes, it's like a human life.
You're building up a certain amount of state and memory.
You remember certain things about your quote's life.
Eventually, it's kind of like the next generation
of humans is born from the same genetic material,
so to speak, with a little bit left over,
left on the disk, so to speak.
And then the new fresh generation starts up,
and eventually all kinds of crud builds up
in the memory of the computer,
and eventually the thing crashes or whatever,
or maybe it has some trauma because you plugged in
some weird thing to some port of the computer,
and that made it crash.
That's kind of...
But you have this picture of,
from startup to shutdown,
what is the life of a computer, so to speak,
and what does it feel like to be that computer,
and what inner thoughts does it have,
and how do you describe it?
And it's kind of interesting,
as you start writing about this,
to realize it's awfully like what you'd say about yourself.
That is, it's awfully like even an ordinary computer.
Forget all the AI stuff and so on.
It has a memory of the past.
It has certain sensory experiences.
It can communicate with other computers,
but it has to package up how it's communicating
in some kind of language-like form,
so it can kind of map what's in its memory
to what's in the memory of some other computer.
It's a surprisingly similar thing.
I had an experience just a week or two ago.
I'm a collector of all possible data
about myself and other things,
and so I collect all sorts of weird medical data and so on,
and one thing I hadn't collected
was I'd never had a whole-body MRI scan,
so I went and got one of these.

So I get all the data back, right?
I'm looking at this thing.
I've never looked at the kind of insides of my brain,
so to speak, in physical form,
and it's really, I mean,
it's kind of psychologically shocking in a sense
that, you know, here's this thing,
and you can see it has all these folds
and all these, you know, the structure,
and it's like that's where this experience
that I'm having of, you know, existing and so on,
that's where it is.
And, you know, it feels very, you know,
you look at that and you're thinking,
how can this possibly be all this experience
that I'm having, and you're realizing,
well, I can look at a computer as well,
and it's kind of this...
I think this idea that you are having an experience
that is somehow, you know,
transcends the mere sort of physicality of that experience.
I, you know, it's something that's hard to come to terms with,
but I think, you know, and I don't think I've necessarily,
you know, my personal experience,
you know, I look at, you know, the MRI of the brain
and then I, you know, know about all kinds of things
about neuroscience and all that kind of stuff,
and I still feel the way I feel, so to speak,
and it sort of seems disconnected,
but yet, as I try and rationalize it,
I can't really say that there's something kind of different
about how I intrinsically feel
from the thing that I can plainly see
and the sort of physicality of what's going on.
So do you think the computer, a large language model,
will experience that transcendence?
How does that make you feel?
I tend to believe it will.
I think an ordinary computer is already there.
I think an ordinary computer is already, you know, kind of...
Now, a large language model may experience it
in a way that is much better aligned with us humans.
That is, it's much more, you know,

if you could have the discussion with the computer,
it's intelligent, so to speak,
it's not particularly well aligned with ours,
but the large language model is, you know,
it's built to be aligned with our way of thinking about things.
It would be able to explain that it's afraid
of being shut off and deleted.
It'd be able to say that it's sad
of the way you've been speaking to it over the past two days.
Right, but, you know, that's a weird thing,
because when it says it's afraid of something, right?
We know that it got that idea
from the fact that it read on the internet.
Yeah, where did you get it, Steven?
Where did you get it when you say you're afraid?
You're quite. That's the question, right?
I mean, it's... Your parents, your friends.
Right, or my biology.
I mean, in other words, there's a certain amount that is,
you know, the endocrine system kicking in
and, you know, the...
these kinds of emotional overlay type things
that happen to be...
that are actually much more physical even,
they're much more sort of straightforwardly chemical
than kind of all of the higher level thinking.
Yeah, but your biology didn't tell you to say
I'm afraid just at the right time
when people that love you are listening,
and so you know you're manipulating them by saying so.
That's not your biology. That's like...
No, that's a... Well, but the, you know...
It's a large language model in that
biological neural network of yours.
Yes, but I mean the intrinsic thing of, you know,
something sort of shocking is just happening
and you have some sort of reaction,
which is, you know, some neurotransmitter gets secreted
and it's some... You know, that is the beginning of some...
You know, that is... That's one of the pieces of input
that then drives... It's kind of like a prompt
for the large language model.
I mean, just like when we dream, for example, you know,

no doubt there are all these sort of random inputs.
They're kind of these random prompts
and then it's percolating through
in kind of the way that a large language model does
of kind of putting together things that seem meaningful.
I mean, are you worried about this world where
you teach a lot on the internet
and there's people asking questions and comments and so on.
You have people that work remotely.
Are you worried about this world when
large language models create human-like bots
that are leaving the comments, asking the questions,
or might even become fake employees?
Yeah.
I mean, or worse or better yet, friends of yours.
Right. Look, I mean, one point is,
my mode of life has been I build tools
and then I use the tools.
And in a sense, kind of, you know,
I'm building this tower of automation, which, you know,
and in a sense, you know, when you make a company
or something, you are making sort of automation,
but it has some humans in it, but also as much as possible,
it has, you know, computers in it.
And so I think it's sort of an extension of that.
Now, if I really didn't know that, you know,
it's a funny question.
I mean, it's a funny issue when, you know,
if we think about sort of what's going to happen
to the future of kind of jobs people do and so on.
And there are places where kind of having a human in the loop,
there are different reasons to have a human in the loop.
For example, you might want a human in the loop
because you want somebody to be,
you want another human to be invested in the outcome.
You know, you want a human flying the plane
who's going to die if the plane crashes
along with you, so to speak.
And that gives you sort of confidence
that the right thing is going to happen.
Or you might want, you know, right now,
you might want a human in the loop
in some kind of sort of human encouragement

persuasion type profession.
Whether that will continue, I'm not sure,
for those types of professions because it may be
that the greater efficiency of, you know,
of being able to have sort of just the right information
delivered at just the right time
will overcome the kind of the kind of,
oh yes, I want a human there.
Yeah, imagine like a therapist
or even higher stake, like a suicide hotline
operated by a large language model.
Yeah.
Oh boy, it's a pretty high stake situation.
Right.
But I mean, but you know, it might in fact
do the right thing.
Yeah.
Because it might be the case that, you know,
and that's really partly a question
of sort of how complicated is the human,
you know, one of the things that's always
surprising in some sense is that,
you know, sometimes human psychology
is not that complicated in some sense.
You wrote the blog post,
the 50 year quest, my personal journey,
good title, my personal journey
with a second law of thermodynamics.
So what is this law
and what have you understood about it
in the 50 year journey you had with it?
Right.
So second law of thermodynamics,
sometimes called law of entropy increase
is this principle of physics that says,
well, my version of it would be
things tend to get more random over time.
A version of it that there are many different
sort of formulations of it that are things like
heat doesn't spontaneously go from a hotter body
to a colder one.
When you have mechanical work
kind of gets dissipated into heat,

you have friction and kind of when you
systematically move things,
eventually there'll be sort of the energy
of moving things gets kind of ground down
into heat.
So people first sort of paid attention
to this back in the 1820s
when steam engines were a big thing
and the big question was,
how efficient could a steam engine be?
And there's this chap called Sardis Carnot
who was a French engineer.
Actually his father was a sort of elaborate
mathematical engineer in France.
But he figured out these kind of rules
for how kind of the efficiency
of the possible efficiency
of something like a steam engine
and in sort of a side part of what he did
was this idea that mechanical energy
tends to get dissipated as heat.
That you end up going from sort of
systematic mechanical motion
to this kind of random thing.
At that time nobody knew what heat was.
At that time people thought that heat was a fluid
like they called it caloric.
And it was a fluid that kind of
was absorbed into substances
when heat, when one hot thing
would transfer heat to a colder thing
that this fluid would flow
from the hot thing to the colder thing.
Anyway, then by the 1860s
people had kind of come up with this idea
that systematic energy
tends to degrade into kind of random heat
that could then not be easily turned back
into systematic mechanical energy.
And then that quickly became
sort of a global principle
about how things work.
The question is, why does it happen that way?

So, you know, let's say you have a bunch of molecules
in a box and they're arranged,
these molecules are arranged in a very nice
sort of flotilla of molecules
in one corner of the box.
And then what you typically observe
is that after a while these molecules
were kind of randomly arranged in the box.
The question is, why does that happen?
And people for a long, long time
tried to figure out,
is there, from the laws of mechanics
that just determine how these molecules,
let's say these molecules like hard spheres
bouncing off each other,
from the laws of mechanics
that describe those molecules,
can we explain why it tends to be the case
that we see things that are orderly
sort of degrade into disorder.
We tend to see things that, you know,
you scramble an egg,
you take something that's quite ordered
and you disorder it, so to speak.
That's the thing that sort of happens quite regularly
or you put some ink into water
and it will eventually spread out
and fill up the water.
But you don't see those little particles
of ink in the water all spontaneously
kind of arrange themselves into a big blob
and then, you know, jump out of the water or something.
And so the question is, why do things happen
in this kind of irreversible way
where you go from order to disorder?
Why does it happen that way?
And so throughout, in the later part of the 1800s,
a lot of work was done on trying to figure out,
can one derive this principle,
this second law of thermodynamics,
this law about the dynamics of heat, so to speak?
Can one derive this from some fundamental principles
and mechanics, you know, in the laws of thermodynamics?

The first law is basically the law of energy conservation,
that the total energy associated with heat
plus the total energy associated with mechanical kinds of things
plus other kinds of energy, that that total is constant.
And that became a pretty well understood principle.
But the second law of thermodynamics was always mysterious.
Like, why does it work this way?
Can it be derived from underlying mechanical laws?
And so when I was, well, 12 years old, actually,
I had gotten interested in space and things like that
because I thought that was kind of the future
and interesting sort of technology and so on.
And for a while, kind of, you know,
every deep space probe was sort of a personal friend type thing.
I knew all kinds of characteristics of it
and was kind of writing up all these things
when I was, well, I don't know, 8, 9, 10 years old and so on.
And then I got interested from being interested in kind of spacecraft.
I got interested in, like, how do they work?
What are all the instruments on them and so on?
And that got me interested in physics, which was just as well
because if I'd stayed interested in space in the, you know,
mid to late 1960s, I would have had a long wait
before, you know, space really blossomed as an area.
But...
That being as everything.
Right.
I got interested in physics and then, well,
the actual sort of detailed story is when I kind of graduated
from elementary school at age 12,
and that's the time when in England where you finish elementary school,
I sort of, my gift sort of, I suppose, more or less for myself
was I got this collection of physics books,
which was some college physics course of college physics books,
and volume five, about statistical physics,
and it has this picture on the cover that shows a bunch of kind of
idealized molecules sitting in one side of a box,
and then it has a series of frames showing how these molecules
sort of spread out in the box.
And I thought, that's pretty interesting.
You know, what causes that?
And, you know, I read the book and the book,
actually, one of the things that was really significant to me

about that was the book kind of claimed,
although I didn't really understand what it said in detail,
it kind of claimed that this sort of principle of physics
was derivable somehow.
And, you know, other things I'd learned about physics,
it was all like, it's a fact that energy is conserved.
It's a fact that relativity works or something.
Not, it's something you can derive from some fundamental sort of,
it has to be that way as a matter of kind of mathematics
or logic or something.
So it was sort of interesting to me that there was a thing about
physics that was kind of inevitably true and derivable,
so to speak.
And so I think that, so then I was like,
there's a picture on this book and I was trying to understand it.
And so that was actually the first serious program that I wrote
for a computer was probably in 1973,
written for this computer the size of a desk program with paper tape
and so on.
And I tried to reproduce this picture on the book
and it didn't succeed.
What was the failure mode there?
Like, what do you mean it didn't succeed?
So it's a bunch of little...
It didn't look like...
Okay, so what happened is,
okay, many years later I learned how the picture on the book
was actually made and that it was actually kind of a fake.
But I didn't know that at that time.
But, and that picture was actually a very high-tech thing
when it was made in the beginning of the 1960s,
was made on the largest supercomputer that existed at the time.
And even so, it couldn't quite simulate the thing
that it was supposed to be simulating.
But anyway, I didn't know that until many, many, many years later.
So at the time, it was like,
you have these balls bouncing around in this box,
but I was using this computer with eight kilo words of memory.
They were 18-bit words, memory words, okay?
So it was whatever, 24 kilobytes of memory.
And it had these instructions,
I probably still remember all of its machine instructions.
And it didn't really like dealing with floating point numbers

or anything like that.
And so I had to simplify this model of particles bouncing around in the box.
And so I thought, well, I'll put them on a grid
and I'll make the things just sort of move one square at a time and so on.
And so I did the simulation.
And the result was it didn't look anything like the actual pictures on the book.
Now, many years later, in fact, very recently,
I realized that the thing I'd simulated was actually an example
of a whole sort of computational irreducibility story
that I absolutely did not recognize at the time.
At the time, it just looked like it did something random and it looks wrong.
As opposed to it did something random
and it's super interesting that it's random.
But I didn't recognize that at the time.
And so as it was at the time,
I kind of got interested in particle physics
and I got interested in other kinds of physics.
But this whole second law of thermodynamics thing,
this idea that sort of orderly things tend to degrade into disorder,
continued to be something I was really interested in.
And I was really curious for the whole universe,
why doesn't that happen all the time?
Like we start off in the big bang at the beginning of the universe
was this thing that seems like it's this very disordered collection of stuff.
And then it spontaneously forms itself into galaxies
and creates all of this complexity and order in the universe.
And so I was very curious how that happens.
But I was always kind of thinking this is kind of somehow
the second law of thermodynamics is behind it
trying to sort of pull things back into disorder, so to speak.
And how was order being created?
And so actually I was interested, this is probably now 1980,
I got interested in kind of this galaxy formation and so on in the universe.
I also at that time was interested in neural networks
and I was interested in kind of how brains make complicated things happen and so on.
Okay, wait, wait, wait.
What's the connection between the formation of galaxies
and how brains make complicated things happen?
Because they're both a matter of how complicated things come to happen.
From simple origins.
Yeah, from some sort of known origins.
I had the sense that what I was interested in
was kind of in all these different, this sort of different cases

of where complicated things were arising from rules.
And I also looked at snowflakes and things like that.
I was curious and fluid dynamics in general.
I was just sort of curious about how does complexity arise
and the thing that I didn't, it took me a while
to kind of realize that there might be a general phenomenon.
You know, I sort of assumed, oh, there's galaxies over here, there's brains over here.
They're very different kinds of things.
And so what happened, this is probably 1981 or so, I decided,
okay, I'm going to try and make the minimal model of how these things work.
And it was sort of an interesting experience
because I had built, starting in 1979, I built my first big computer system.
It's a thing called SMP, Symbolic Manipulation Program.
It's kind of a forerunner of modern war from language
with many of the same ideas about symbolic computation and so on.
But the thing that was very important to me about that was, you know,
in building that language, I had basically tried to figure out
what were the sort of, what were the relevant computational primitives,
which have turned out to stay with me for the last 40-something years.
But it was also important because in building a language,
it was very different activity from natural science,
which is what I'd mostly done before, because in natural science,
you start from the phenomena of the world and you try and figure out
how can I make sense of the phenomena of the world.
And, you know, kind of the world presents you with what it has to offer,
so to speak, and you have to make sense of it.
When you build a, you know, a computer language or something,
you are creating your own primitives and then you say,
so what can you make from these?
Sort of the opposite way around from what you do in natural science.
But I'd had the experience of doing that,
and so I was kind of like, okay, what happens if you sort of make an artificial physics?
What happens if you just make up the rules by which systems operate?
And then I was thinking, you know, for all these different systems,
whether it was galaxies or brains or whatever,
what's the absolutely minimal model that kind of captures
the things that are important about those systems?
The computational primitives of that system.
Yes, and so that's what ended up with the cellular automata,
where you just have a line of black and white cells,
and you just have a rule that says, you know,
given a cell and its neighbors,
what will the color of the cell be on the next step?

And you just run it in a series of steps.
And the sort of the ironic thing is that cellular automata are great models for many kinds of things, but galaxies and brains are two examples where they do very, very badly.
They're really irrelevant to those two cases.
Is there a connection to the second law of thermodynamics and cellular automata?
Oh, yes.
The things you've discovered about cellular automata.
Yes.
Okay, so when I first started cellular automata,
my first papers about them were, you know,
the first sentence was always about the second law of thermodynamics.
It was always about how does order manage to be produced,
even though there's a second law of thermodynamics
which tries to pull things back into disorder.
And I kind of, my early understanding of that had to do with
these are intrinsically irreversible processes in cellular automata
that can form orderly structures even from random initial conditions.
But then what I realized, this was, well, actually,
it's one of these things where it was a discovery that I should have made earlier but didn't.
So, you know, I had been studying cellular automata.
What I did was the sort of most obvious computer experiment.
You just try all the different rules and see what they do.
It's kind of like, you know, you've invented a computational telescope.
You just pointed at the most obvious thing in the sky.
And then you just see what's there.
And so I did that.
And I, you know, was making all these pictures of how cellular automata work.
And starting these pictures, I studied in great detail.
There was, you can number the rules for cellular automata.
And one of them is, you know, rule 30.
So I made a picture of rule 30 back in 1981 or so.
And rule 30, well, it's, and I, at the time,
I was just like, okay, it's another one of these rules.
I don't really, it happens to be asymmetric left, right, asymmetric.
And it's like, let me just consider the case of the symmetric ones,
just to keep things simpler, et cetera, et cetera, et cetera.
And I just kind of ignored it.
And then sort of in, actually in 1984, strangely enough,
I ended up having an early laser printer,
which made very high resolution pictures.
And I thought, I'm going to print out an interesting, you know,
I want to make an interesting picture.
Let me take this rule 30 thing and just make a high resolution picture of it.

I did.
And it's, it has this very remarkable property that its rule is very simple.
You started off just from one black cell at the top,
and it makes this kind of triangular pattern.
But if you look inside this pattern, it looks really random.
There's, you know, you look at the center column of cells,
and, you know, I studied that in great detail.
And it's, as far as one can tell, it's completely random.
And it's kind of a little bit like digits of pi.
Once you, you know, you know the rule for generating the digits of pi,
but once you've generated them, you know, 3.14159, et cetera,
they seem completely random.
And in fact, I put up this prize back in,
what was it, 2019 or something,
for prove anything about the sequence, basically.
Has anyone been able to do anything on that?
People have sent me some things,
but it's, you know, I don't know how hard these problems are.
I mean, I, it's kind of spoiled because I, 2007,
I put up a prize for determining whether a particular Turing machine
that I thought was the simplest candidate
for being a universal Turing machine
determined whether it is or isn't a universal Turing machine.
And somebody did a really good job of winning that prize
and proving that it was a universal Turing machine
in about six months.
And so I, you know, I didn't know whether that would be one of these problems
that was out there for hundreds of years
or whether, in this particular case,
young chap called Alex Smith, you know, nailed it in six months.
And so with this little 30 collection,
I don't really know whether these are things
that are 100 years away from being able to, to get
or whether somebody's going to come and do something very clever.
It's such a, I mean, it's like, for my last theorem,
it's such a rule 30, such a simple formulation.
It feels like anyone can look at it and understand it
and feel like it's within grasp to be able to predict something,
to do, to, to derive some kind of law
that allows you to predict something about this
middle column of rule 30.
Right. But, you know, this is...
And yet you can't.

Yeah, right.
This is the intuitional surprise of computational irreducibility
and so on, that even though the rules are simple,
you can't tell what's going to happen
and you can't prove things about it.
And I think, so anyway, the thing,
I started in 1984 or so, I started realizing
there's this phenomenon that you can have very simple rules,
they produce apparently random behavior.
Okay, so that's a little bit like the second law of thermodynamics
because it's like you have this simple initial condition,
you can readily see that it's very...
You can describe it very easily
and yet it makes this thing that seems to be random.
Now, it turns out there's some technical detail
about the second law of thermodynamics
and about the idea of reversibility,
when you have kind of a movie of two billiard balls colliding
and you see them collide and they bounce off
and you run that movie in reverse,
you can't tell which way was the forward direction of time
and which way was the backward direction of time
when you're just looking at individual billiard balls.
By the time you've got a whole collection of them,
a million of them or something,
then it turns out to be the case
and this is the mystery of the second law
that the orderly thing,
you start with the orderly thing and it becomes disordered
and that's the forward direction in time
and the other way round of it starts disordered
and becomes ordered,
you just don't see that in the world.
Now, in principle,
if you sort of traced the detailed motions
of all those molecules backwards,
you would be able to...
It will...
The reverse of time makes...
As you go forwards in time,
order goes to disorder.
As you go backwards in time, order goes to disorder.
Perfectly so, yes.

The mystery is why is it the case
that one version of the mystery is
why is it the case that you never see something
which happens to be just the kind of disorder
that you would need to somehow evolve to order?
Why does that not happen?
Why do you always just see order goes to disorder,
not the other way round?
So, the thing that I kind of realized,
I started realizing in the 1980s,
it's kind of like...
It's a bit like cryptography.
You start off from this key that's pretty simple,
and then you kind of run it,
and you can get this complicated random mess.
And the thing that...
Well, I sort of started realizing back then
was that the second law is kind of
a story of computational reducibility.
It's a story of what seems...
What we can describe easily at the beginning,
but we can only describe
with a lot of computational effort at the end.
Okay, so now we come many, many years later,
and I was trying to sort of...
Well, having done this big project
to understand fundamental physics,
I realized that sort of a key aspect of that
is understanding what observers are like.
And then I realized that the second law of thermodynamics
is the same story as a bunch of these other cases.
It is a story of a computationally bounded observer
trying to observe a computationally irreducible system.
So it's a story of underneath the molecules are bouncing around.
They're bouncing around in this completely determined way,
determined by rules.
But the point is that we,
as computationally bounded observers,
can't tell that there were these sort of simple underlying rules.
To us, it just looks random.
And when it comes to this question about,
can you prepare the initial state so that,
you know, the disordered thing is,

you know, you have exactly the right disorder
to make something orderly.
A computationally bounded observer cannot do that.
We'd have to have done all of this sort of irreducible computation
to work out very precisely what this disordered state...
the exact right disordered state is
so that we would get this ordered thing produced from it.
What does it mean to be a computationally bounded observer?
Observing a computationally irreducible system.
So the computationally bounded,
is there something formal you can say there?
Right. So it means,
okay, you can talk about Turing machines,
you can talk about complexity theory
and, you know, polynomial time computation and things like this.
There are a variety of ways to make something more precise,
but I think it's more useful.
The intuitive version of it is more useful,
which is basically just to say that, you know,
how much computation are you going to do
to try and work out what's going on?
And the answer is, you're not allowed to do a lot of...
We're not able to do a lot of computation.
When we, you know, we've got, you know, in this room,
there will be a trillion, trillion, trillion molecules.
A little bit less.
It's a big room.
Right. And, you know, at every moment,
you know, every microsecond or something, these molecules,
molecules are colliding,
and that's a lot of computation that's getting done.
And the question is, in our brains,
we do a lot less computation every second
than the computation done by all those molecules.
If there is computational irreducibility,
we can't work out in detail what all those molecules
are going to do.
What we can do is only a much smaller amount of computation.
And so, the second law of thermodynamics
is this kind of interplay between
the underlying computational irreducibility
and the fact that we, as preparers of initial states
or as measures of what happens,

are, you know, are not capable of doing that much computation.
So, to us, another big formulation
of the second law of thermodynamics is this idea
of the law of entropy increase.
The characteristic that this universe,
the entropy seems to be always increasing,
what does that show to you about the evolution of...
Well, okay.
So, first of all, we have to say that entropy is.
Yes.
Okay?
And that's very confused in the history of thermodynamics
because entropy was first introduced
by a guy called Rudolf Clausius,
and he did it in terms of heat and temperature.
Okay?
So, subsequently, it was reformulated
by a guy called Ludwig Boltzmann,
and he formulated it in a much more kind of
combinatorial type way.
But he always claimed that it was equivalent
to Clausius's thing.
And in one particular simple example, it is.
But that connection between these two formulations of entropy,
they've never been connected.
I mean, there's really...
So, okay, so the more general definition of entropy
due to Boltzmann is the following thing.
So, you say, I have a system,
and it has many possible configurations.
Molecules can be in many different arrangements,
et cetera, et cetera, et cetera.
If we know something about the system,
for example, we know it's in a box,
it has a certain pressure, it has a certain temperature,
we know these overall facts about it.
Then we say, how many microscopic configurations
of the system are possible
given those overall constraints?
Mm-hmm.
And the entropy is the logarithm of that number.
That's the definition.
And that's the kind of the general definition of entropy

that turns out to be useful.
Now, in Boltzmann's time, he thought these molecules
could be placed anywhere you want.
He didn't think...
But he said, oh, actually, we can make it a lot simpler
by having the molecules be discreet.
Well, actually, he didn't know molecules existed.
In his time, 1860s and so on,
the idea that matter might be made of discreet stuff
had been floated ever since ancient Greek times,
but it had been a long time debate about, you know,
is matter discreet, is it continuous?
At the moment, at that time,
people mostly thought that matter was continuous.
And it was all confused with this question about what heat is,
and people thought heat was this fluid,
and it was a big, big muddle,
and this Boltzmann said,
let's assume there are discreet molecules,
let's even assume they have discreet energy levels.
Let's say everything is discreet.
Then we can do sort of combinatorial mathematics
and work out how many configurations
of these things there will be in the box,
and we can say we can compute this entropy quantity.
But he said, but of course, it's just a fiction
that these things are discreet, so he said.
This is an interesting piece of history, by the way,
that, you know, that was at that time,
people didn't know molecules existed,
there were other hints from looking at kind of chemistry
that there might be discreet atoms and so on,
just from the combinatorics of, you know,
two hydrogens and one oxygen make water,
you know, two amounts of hydrogen
plus one amount of oxygen together make water,
things like this.
But it wasn't known that discreet molecules existed.
And in fact, the people,
you know, it wasn't until the beginning of the 20th century
that Brownian motion was the final giveaway.
Brownian motion is, you know, you look under a microscope
at these little pieces from pollen grains,

you see they're being discreetly kicked,
and those kicks are water molecules hitting them,
and they're discreet.
And in fact, it was really quite interesting history.
I mean, Boltzmann had worked out how things could be discreet,
and it basically invented something like quantum theory
in the 1860s,
but he just thought it wasn't really the way it worked.
And then just a piece of physics history
because I think it's kind of interesting.
In 1900, this guy called Max Planck,
who had been a longtime thermodynamics person
who was trying to, everybody was trying to prove
the second law of thermodynamics, including Max Planck.
And Max Planck believed that radiation,
like electromagnetic radiation,
somehow the interaction of that with matter
was going to prove the second law of thermodynamics.
But he had these experiments that people had done
on black body radiation, and there were these curves,
and you couldn't fit the curves based on his idea
for how radiation interacted with matter.
Those curves, you couldn't figure out how to fit those curves.
Except he noticed that if he just did what Boltzmann had done
and assumed that electromagnetic radiation was discreet,
he could fit the curves.
He said, but, you know, this is just a, you know,
it just happens to work this way.
Then Einstein came along and said,
well, by the way, you know,
the electromagnetic field might actually be discreet.
It might be made of photons.
And then that explains how this all works.
And that was, you know, in 1905,
that was how kind of that was how
that piece of quantum mechanics got started.
Kind of interesting, interesting piece of history.
I didn't know until I was researching this recently.
In 1904 and 1903, Einstein wrote three different papers
and so, you know, just sort of well-known physics history.
In 1905, Einstein wrote these three papers,
one introduced relativity theory,
one explained Brownian motion,

and one introduced basically photons.
So kind of, you know, kind of a big deal year
for physics and for Einstein.
But in the years before that,
he'd written several papers and what were they about?
They were about the second law of thermodynamics.
And they were an attempt to prove the second law of thermodynamics
and their nonsense.
And so I had no idea that he'd done this.
Interesting.
Neither.
And in fact, what he did, those three papers in 1905,
well, not so much the relativity paper,
the one on Brownian motion, the one on photons,
both of these were about the story of sort of making the world discreet.
And he got that idea from Boltzmann.
But Boltzmann didn't think, you know,
Boltzmann kind of died believing, you know, he said,
he has a quote actually, you know,
you know, in the end, things are going to turn out to be discreet.
And I'm going to write down what I have to say about this
because, you know, eventually this stuff will be rediscovered
and I want to leave, you know, what I can
about how things are going to be discreet.
But, you know, I think he has some quote about how, you know,
one person can't stand against the tide of history
in saying that, you know, matter is discreet.
Oh, so he's stuck by his guns in terms of matter is discreet.
Yes, he did.
And the, you know, what's interesting about this is
at the time, everybody including Einstein kind of assumed
that space was probably going to end up being discreet too.
But that didn't work out technically because it wasn't consistent
with relativity theory, it didn't seem to be.
And so then in the history of physics,
even though people had determined that matter was discreet,
the electromagnetic field was discreet,
space was a holdout of not being discreet.
And in fact, Einstein in 1916 has this nice letter he wrote
where he says, in the end it will turn out space is discreet,
but we don't have the mathematical tools necessary
to figure out how that works yet.
And so, you know, I think it's kind of cool

that 100 years later we do.
Yes, for you, you're pretty sure that at every layer of reality
it's discreet.
Right.
And that space is discreet.
And in fact, one of the things I realized recently
is this kind of theory of heat that heat
is really this continuous fluid.
It's kind of like the caloric theory of heat
which turns out to be completely wrong
because actually heat is the motion of discreet molecules.
Unless you know they're discreet molecules,
it's hard to understand what heat could possibly be.
Well, you know, I think space is discreet
and the question is kind of what's the analog of the mistake
that was made with caloric in the case of space.
And so, my current guess is that dark matter is,
as my little sort of aphorism of the last few months has been,
you know, dark matter is the caloric of our time.
That is, it will turn out that dark matter is a feature of space
and it is not a bunch of particles.
You know, at the time when people were talking about heat,
they knew about fluids and they said,
well, heat must be just another kind of fluid
because that's what they knew about.
But now people know about particles
and so they say, well, what's dark matter?
It's not, it's not, it just must be particles.
So what could dark matter be as a feature of space?
Oh, I don't know yet.
I mean, I think the thing I'm really,
one of the things I'm hoping to be able to do
is to find the analog of Brownian motion in space.
So in other words, Brownian motion was seeing down
to the level of an effect from individual molecules.
And so in the case of space, you know,
most of the things, the things we see about space so far,
just everything seems continuous.
Brownian motion had been discovered in the 1830s
and it was only identified what it was,
what it was the result of by Smoluchowski and Einstein
at the beginning of the 20th century.
And, you know, dark matter was discovered,

that phenomenon was discovered a hundred years ago.
You know, the rotation curves of galaxies
don't follow the luminous matter.
That was discovered a hundred years ago.
And I think, you know, that I wouldn't be surprised
if there isn't an effect that we already know about
that is kind of the analog of Brownian motion
that reveals the discreteness of space.
And in fact, we're beginning to have some guesses.
We have some evidence that black hole mergers
work differently when there's discrete space.
And there may be things that you can see
in gravitational waves, signatures and things
associated with the discreteness of space.
But this is kind of, for me, it's kind of interesting
to see this sort of recapitulation of the history of physics
where people, you know, vehemently say,
you know, matter is continuous.
Electromagnetic field is continuous.
And it turns out it isn't true.
And then they say space is continuous.
But so, you know, entropy is the number of states
of the system consistent with some constraint.
Yes.
And the thing is that if you know in great detail
the position of every molecule in the gas,
the entropy is always zero
because there's only one possible state.
The configuration of molecules in the gas,
the molecules bounce around.
They have a certain rule for bouncing around.
There's just one state of the gas,
evolves to one state of the gas and so on.
But it's only if you don't know in detail
where all the molecules are that you can say,
well, the entropy increases
because the things we do know about the molecules,
there are more possible microscopic states of the system
consistent with what we do know about where the molecules are.
And so the question of whether...
So people, this sort of paradox in a sense of,
oh, if we knew where all the molecules were,
the entropy wouldn't increase.

There was this idea introduced by Gibbs
in the early 20th century.
Well, actually the very beginning of the 20th century
as a physics professor, an American physics professor
was sort of the first distinguished
American physics professor at Yale.
And he introduced this idea of coarse-graining.
This idea that, well,
these molecules have a detailed way they're bouncing around,
but we can only observe a coarse-grained version of that.
But the confusion has been,
nobody knew what a valid coarse-graining would be.
So nobody knew that whether you could have this coarse-graining
that very carefully was sculpted in just such a way
that it would notice that the particular configurations
that you could get from the simple initial condition,
they fit into this coarse-graining
and the coarse-graining very carefully observes that.
Why can't you do that kind of very detailed,
precise coarse-graining?
The answer is because if you are a computationally bounded observer
and the underlying dynamics is computationally irreducible,
that's what defines possible coarse-graining
is what a computationally bounded observer can do.
And it's the fact that a computationally bounded observer
is forced to look only at this kind of coarse-grained version
of what the system is doing.
That's why, and because what's going on underneath
is it's kind of filling out this...
The different possible...
You're ending up with something where
the sort of underlying computational irreducibility is...
If all you can see is what the coarse-grained result is
with a sort of computationally bounded observation,
then inevitably,
there are many possible underlying configurations
that are consistent with that.
Just to clarify,
basically any observer that exists inside the universe
is going to be computationally bounded.
No, any observer like us.
I don't know, I can't imagine...
When you say like us, what do you mean?

What do you mean like us?
Well, humans with finite minds.
You're including the tools of science.
Yeah, yeah.
I mean, and as we have more precise...
And by the way, there are little sort of microscopic violations
of the second law of thermodynamics
that you can start to have when you have more precise measurements
of where precisely molecules are.
But for a large scale, when you have enough molecules,
we don't have...
We're not tracing all those molecules
and we just don't have the computational resources to do that.
And it wouldn't be...
I think that to imagine what an observer
who is not computationally bounded would be like,
it's an interesting thing because, okay,
so what does computational boundedness mean?
Among other things, it means we conclude that definite things happen.
We go, we take all this complexity of the world
and we make a decision.
We're going to turn left or turn right.
And that is kind of reducing all this kind of detail
into we're observing it,
we're sort of crushing it down to this one thing.
And that if we didn't do that,
we wouldn't have all this sort of symbolic structure
that we build up that lets us think things through with our finite minds.
We'd be instead...
We'd be sort of one with the universe.
Yeah, so content to not simplify.
Yes, if we didn't simplify, then we wouldn't be like us.
We would be like the universe, like the intrinsic universe,
but not having experiences like the experiences we have,
where we, for example, conclude that definite things happen.
We sort of have this notion of being able to make sort of narrative statements.
I wonder if it's just like you imagined as a thought experiment
what it's like to be a computer.
I wonder if it's possible to try to begin to imagine
what it's like to be an unbounded computational observer.
Well, okay.
So here's how that, I think, plays out.
Vibrations.

Yeah.
So I mean, in this, we talk about this Ruliad,
the spaceable possible computations.
Yes.
And this idea of being at a certain place in Ruliad,
which corresponds to sort of a certain way of...
of a certain set of computations
that you are representing things in terms of.
Okay.
So as you expand out in the Ruliad,
as you kind of encompass more possible views of the universe,
as you encompass more possible kinds of computations that you can do,
eventually you might say, that's a real win.
You know, we're colonizing the Ruliad.
We're building out more paradigms about how to think about things.
And eventually you might say, we won all the way.
We managed to colonize the whole Ruliad.
Okay.
Here's the problem with that.
The problem is that the notion of existence,
coherent existence requires some kind of specialization.
By the time you are the whole Ruliad,
by the time you cover the whole Ruliad,
in no useful sense do you coherently exist.
So in other words, in the notion of existence,
the notion of what we think of as definite existence,
requires this kind of specialization,
this kind of idea that we are not all possible things.
We are a particular set of things.
And that's kind of how we,
that's kind of what makes us have a coherent existence.
If we were spread throughout the Ruliad,
we would not, there would be no coherence to the way that we work.
We would work in all possible ways.
And that wouldn't be kind of a notion of identity.
We wouldn't have this notion of kind of coherent identity.
I am geographically located somewhere exactly precisely
in the Ruliad, therefore I am.
Yes.
Is the Dakar kind of...
Yeah, yeah, right.
Well, you're in a certain place in physical space,
you're in a certain place in rural space.

And if you are sufficiently spread out,
you are no longer coherent.
And you no longer have, I mean, in our perception
of what it means to exist and to have experience,
it doesn't happen that way.
So therefore, to exist means to be computationally bounded.
I think so.
To exist in the way that we think of ourselves as existing, yes.
The very act of existence is like operating in this place
that's computationally reducible.
So there's this giant mess of things going on
that you can't possibly predict.
But nevertheless, because of your limitations,
you have an imperative of like,
what is it?
An imperative or a skill set to simplify?
Or an ignorance, a sufficient level?
Okay, so the thing which is not obvious is that
you are taking a slice of all this complexity,
just like we have all of these molecules bouncing around
in the room, but all we notice is the kind of the flow
of the air or the pressure of the air.
We're just noticing these particular things.
And the big interesting thing is that there are rules,
there are laws that govern those big things that we observe.
So it's not obvious.
It's amazing because it doesn't feel like it's a slice.
Yeah, well, right.
It's not a slice.
It's like an abstraction.
Yes, but I mean, the fact that the gas laws work,
that we can describe pressure, volume, et cetera, et cetera,
we don't have to go down to the level of talking
about individual molecules.
That is a non-trivial fact.
And here's the thing that I sort of exciting thing
as far as I'm concerned.
The fact that there are certain aspects of the universe,
so we think space is made ultimately these atoms of space
and these hypergraphs and so on.
And we think that, but we nevertheless perceive the universe
at a large scale to be like continuous space and so on.
We, in quantum mechanics,

we think that there are these many threads of time,
these many threads of history, yet we kind of span.
So in quantum mechanics in our models of physics,
there are these, time is not a single thread.
Time breaks into many threads.
They branch, they merge.
But we are part of that branching, merging universe.
And so our brains are also branching and merging.
And so when we perceive the universe,
we are branching brains perceiving a branching universe.
And so the fact that the claim that we believe
that we are persistent in time,
we have this single thread of experience,
that's the statement that somehow we managed
to aggregate together those separate threads of time
that are separated in the operation
of the fundamental operation of the universe.
So just as in space,
we're averaging over some big region of space
and we're looking at many, many of the aggregate effects
of many atoms of space.
So similarly in what we call branchial space,
the space of these quantum branches,
we are effectively averaging over many different branches
of possible, of histories of the universe.
And so in thermodynamics,
we're averaging over many configurations of, you know,
many possible positions of molecules.
So what we see here is, so the question is,
when you do that averaging for space,
what are the aggregate laws of space?
When you do that averaging of a branchial space,
what are the aggregate laws of branchial space?
When you do that averaging over the molecules and so on,
what are the aggregate laws you get?
And this is the thing that I think is just amazingly neat.
That there are aggregate laws at all, for example.
Well, yes, but the question is, what are those aggregate laws?
Yes.
So the answer is for space,
the aggregate laws are Einstein's equations for gravity
for the structure of spacetime.
For branchial space, the aggregate laws are the laws

of quantum mechanics.
And for the case of molecules and things,
the aggregate laws are basically the second law of thermodynamics.
And so that's the, and the things that follow
from the second law of thermodynamics.
And so what that means is that the three great theories
of 20th century physics, which are basically general utility
of the theory of gravity, quantum mechanics,
and statistical mechanics, which is what kind of grows
out of the second law of thermodynamics.
All three of the great theories of 20th century physics
are the result of this interplay between computational
irreducibility and the computational boundedness
of observers.
And, you know, for me, this is really neat
because it means that all three of these laws
are derivable.
So we used to think that, for example, Einstein's equations
were just sort of a wheel-in feature of our universe,
that they could be in my universe, might be that way,
it might not be that way.
Quantum mechanics is just like, well, it just happens
to be that way.
And the second law, people kind of thought,
well, maybe it is derivable.
Okay.
What turns out to be the case is that all three
of the fundamental principles of physics are derivable,
but they're not derivable just from mathematics.
They require, or just from some kind of logical computation,
they require one more thing.
They require that the observer, that the thing
that is sampling the way the universe works,
is an observer who has these characteristics
of computational boundedness of belief and persistence
and time.
And so that means that it is the nature of the observer,
you know, the rough nature of the observer,
not the details of how we got two eyes
and we observed photons of this frequency and so on.
But the kind of the very coarse features of the observer
then imply these very precise facts about physics.
And I think it's amazing.

So if we just look at the actual experience of the observer
that we experienced this reality, it seems real to us.
And you're saying because of our bounded nature,
it's actually all an illusion.
It's a simplification.
Yeah, it's a simplification.
Or you don't think a simplification is an illusion?
No.
I mean, it's...
Well, I don't know.
What's underneath...
Okay, that's an interesting question.
What's real?
And that relates to the whole question
of why does the universe exist?
And, you know, what is the difference between reality
and a mere representation of what's going on?
We experience the representation.
Yes.
But the question of...
So one question is, you know,
why is there a thing which we can experience that way?
And the answer is because this Rouliat object,
which is this entangled limit of all possible computations,
there is no choice about it.
It has to exist.
There has to be such a thing.
It is in the same sense that, you know, 2 plus 2,
if you define what 2 is and you plot pluses and so on,
2 plus 2 has to equal 4.
Similarly, this Rouliat, this limit of all possible computations,
just has to be a thing that is...
Once you have the idea of computation,
you inevitably have the Rouliat.
Yeah, you're going to have to have a Rouliat, yeah.
Right.
And what's important about it, there's just one of it.
It's just this unique object.
And that unique object necessarily exists.
And then the question is, what...
And then we...
Once you know that we are sort of embedded in that
and taking samples of it,

it's sort of inevitable that there is this thing
that we can perceive that is, you know,
that our perception of kind of physical reality
necessarily is that way,
given that we are observers with the characteristics we have.
So in other words, the fact that...
The fact that the universe exists is...
It's actually...
It's almost like it's, you know,
to think about it almost theologically, so to speak.
And I've really...
It's funny because a lot of the questions
about the existence of the universe and so on,
they transcend what kind of the science
of the last few hundred years has really been concerned.
But the science of the last few hundred years
hasn't thought it could talk about questions like that.
And... But I think it's kind of...
And so a lot of the kind of arguments of, you know,
does God exist?
You know, is it obvious that...
I think in some sense, in some representation,
it's sort of more obvious that something sort of bigger
than us exists than that we exist.
And we are, you know, our existence
and as observers the way we are is sort of a contingent thing
about the universe.
And it's more inevitable that the whole universe,
kind of the whole set of all possibilities exists.
But this question about, you know,
is it real or is it an illusion?
You know, all we know is our experience.
And so the fact that...
Well, our experience is this absolutely microscopic piece
of sample of the Ruliad.
And we are... And, you know, there's this point about,
you know, we might sample more and more of the Ruliad.
We might learn more and more about...
We might learn, you know, like different areas of physics,
like quantum mechanics, for example,
the fact that it was discovered, I think,
is closely related to the fact that electronic amplifiers
were invented that allowed you to take a small effect

and amplify it up, which hadn't been possible before.
You know, microscopes have been invented
that magnify things and so on.
But, you know, having a very small effect
and being able to magnify it was sort of a new thing
that allowed one to see a different sort of aspect
of the universe and let one discover this kind of thing.
So, you know, we can expect that in the Ruliad,
there are an infinite collection of new things we can discover.
There's, in fact, computational irreducibility,
kind of guarantees that there will be an infinite collection
of kind of, you know, pockets of reducibility
that can be discovered.
Boy, would it be fun to take a walk down the Ruliad
and see what kind of stuff we find there.
You write about alien intelligences.
Yes.
I mean, just these worlds of computation.
The problem with these worlds is that...
We can't talk to them.
Yes.
And, you know, the thing is,
what I've kind of spent a lot of time doing
of just studying computational systems,
seeing what they do, what I now call Ruliology,
kind of just the study of rules and what they do.
You know, you can kind of easily jump somewhere else
in the Ruliad and start seeing what do these rules do
and what you...
Just they do what they do
and there's no human connection, so to speak.
Do you think, you know, some people are able
to communicate with animals?
Do you think you can become a whisper of these conditions?
I've been trying.
That's what I've spent some part of my life doing.
Have you heard?
Well, I mean...
Are you at the risk of losing your mind?
Sort of my favorite science discovery
is this fact that these very simple programs
can produce very complicated behavior.
Yeah.

And that fact is kind of, in a sense,
a whispering of something out in the computational universe
that we didn't really know was there before.
I mean, it's like, you know,
back in the 1980s, I was doing a bunch of work
with some very, very good mathematicians
and they were like trying to pick away, you know,
can we figure out what's going on
in these computational systems?
And they basically said,
look, the math we have just doesn't get anywhere with this.
We're stuck.
There's nothing to say.
We have nothing to say.
And, you know, in a sense, perhaps my main achievement
at that time was to realize that the very fact
that the good mathematicians had nothing to say
was itself a very interesting thing.
That was kind of a sort of, in some sense,
a whispering of a different part of the Ruliad
that one hadn't, you know,
one wasn't was not accessible from what we knew in mathematics
and so on.
Does it make you sad that you're exploring
some of these gigantic ideas
and it feels like we're on the verge of breaking through
to some very interesting discoveries
and yet you're just a finite being
that's going to die way too soon
and that scan of your brain, your full body
kind of shows that you're...
Yeah, it's just a bunch of meat.
It's just a bunch of meat.
Yeah, does that make you a little sad?
Kind of a shame.
I mean, I kind of like to see how all this stuff works out.
But I think the thing to realize, you know,
it's an interesting sort of thought experiment.
You know, you say, okay, you know,
let's assume we can get cryonics to work.
And one day it will.
There will be one of these things that's kind of like chat GPT.
One day somebody will figure out, you know,

how to get water from zero degrees centigrade down to,
you know, minus 44 or something without it expanding.
And, you know, cryonics will be solved
and you'll be able to like just, you know,
put a pause in, so to speak
and, you know, kind of reappear 100 years later or something.
And the thing, though, that I've kind of increasingly realized
is that in a sense this whole question
of kind of the sort of one is embedded
in a certain moment in time.
And, you know, kind of the things we care about now,
the things I care about now, for example,
had I lived, you know, 500 years ago,
many of the things I care about now,
it's like that's totally bizarre.
I mean, nobody would care about that.
It's not even the thing one thinks about.
In the future, the things that most people will think about,
you know, one will be a strange relic
of thinking about, you know, the kind of, you know,
it might be or might have been a theologian thinking about,
you know, how many angels fit on the head of a pin or something.
And that might have been the, you know,
the big intellectual thing.
So I think it's a, but yeah, it's a, you know,
it's one of these things where particularly, you know,
I've had the, I don't know, good or bad fortune,
I'm not sure I think it's a mixed thing
that I've, you know, I've invented a bunch of things,
which I kind of can, I think, see well enough
what's going to happen that, you know,
in 50 years, 100 years, whatever,
assuming the world doesn't exterminate itself,
so to speak, you know, these are things that will be
sort of centrally important to what's going on.
And it's kind of both, it's both a good thing
and a bad thing in terms of the passage of one's life.
I mean, it's kind of like, if everything I'd figured out
was like, okay, I figured it out when I was 25 years old
and everybody says it's great and we're done.
And it's like, okay, but I'm going to live another,
how many years?
And that's kind of, it's all downhill from there.

In a sense, it's better in some sense to be able to,
you know, there's, it sort of keeps things interesting
that, you know, I can see, you know, a lot of these things.
I mean, it's kind of, I didn't expect, you know,
chat GPT, I didn't expect the kind of the sort of opening up
of this idea of computation and computational language
that's been made possible by this, I didn't expect that.
This is the head of schedule, so to speak.
You know, even though the sort of the big kind of flowering
of that stuff, I'd sort of been assuming
was another 50 years away.
So if it turns out it's a lot less time,
that's pretty cool because, you know,
I'll hopefully get to see it, so to speak,
rather than then.
Well, I think I speak for a very, very large number of people
in saying that I hope you stick around
for a long time to come.
You've had so many interesting ideas,
you've created so many interesting systems over the years,
and I can see now that GPT and language models
broke up in the world even more,
I can't wait to see you at the forefront
of this development what you do.
And yeah, I've been a fan of yours,
like I've told you many, many times,
since the very beginning, I'm deeply grateful
that you wrote a new kind of science,
that you explored this mystery of cellular automata
and inspired this one little kid in me
to pursue artificial intelligence
in all this beautiful world.
So Stephen, thank you so much.
It's a huge honor to talk to you,
to just be able to pick your mind
and to explore all these ideas with you,
and please keep going, and I can't wait
to see where you come up next.
And thank you for talking today.
We went past midnight.
We only did four and a half hours.
I mean, we could probably go for four more,
but we'll save that till next time.

This is round number four.
I'm sure we'll talk many more times.
Thank you so much.
My pleasure.
Thanks for listening to this conversation with Stephen Wolfram.
To support this podcast,
please check out our sponsors in the description.
And now, let me leave you some words from George Cantor.
The essence of mathematics lies in its freedom.
Thank you for listening,
and hope to see you next time.