Hello everyone. Uh, welcome to another episode of Q&A about history of science
and technology. So, let's see. I think we have a number of questions
um, saved up here. Uh,
right. Well, Graeme is asking, "Just as language has developed and changed throughout history
with new languages,
dialects, and accents, do you think Wolfram Language will grow in that way uh to um be more
regional and categorical
like math versus physics and so on? That's not the goal and that's not what's actually happened
over the last
38 years or so. Kind of our the overall objective is to have kind of a
computational infrastructure for describing the world and to do that in a way that is uh kind of
uniform and
coherent. I think it's it's interesting to see some of the things that have happened. For example,
there are
concepts that back in 1986 1987 when I
was first designing Wolfram Language, there were concepts that were very hard for people to
understand, things about
functional programming, things that related to even representations
of uh different kinds of things with images and things like this. And as time
progressed, those things became more familiar and they became things that were realistic to put
into the
language. I mean, kind of what we're what we're interested in doing is taking sort of the raw
capabilities of
computation and making a bridge between those and uh and what um and what exists
uh and what what we humans can actually think about. One of the things that really helps us is
the human language
like English for example that we get to use human language as a way to explain
what the functionality of Wolfram Language actually is. So for example we
can say something like NestList and there's a notion of nesting and there's
an idea of a list and people already know what those concepts are. If we had to call NestList uh
you know function
373 it would be much harder to understand what was going on. People would not be able to do it.
Sort of an interesting
tale along these lines. Mathematical notation invented what 4–500 years ago,
mostly plus signs, equal signs, things like this. Uh those came in quite
gradually. Those things were sort of introduced you know over over a long
period of time. There was an attempt to make kind of a math-notation-like play
for computer languages done by a chap called Ken Iverson in a language called APL that kind of
originated at the
beginning of the 1960s. And Ken Iverson told me his his goal was to have things
analogous to plus signs and equal signs and so on in the computer language and
to do that with notation. So for example, he had kind of this whole
special keyboard with special characters with kind of triangles and lines through them and and
uh you know things like row-
like characters and iota-like characters and so on. It was really a thing where they were kind of
um uh notational

glyphs that represented computational kinds of things. It was not a success. People didn't manage to understand 50 new glyphs that represented different kinds of computational operations. Now some of those operations, things like for example what's Range in Wolfram Language, was iota or an iota-like character in APL and uh I think oh gosh what was the Map operation I don't remember what it was. The thing that that turned out to be really valuable in Wolfram Language was leveraging the fact that people already know a term like range or like map. They already know those those terms and therefore when they show up in our language it's clear what they mean. I have to say when I built SMP, forerunner to Mathematica and Wolfram Language, I built it starting in 1979. In those days one of the kind of practical constraints was typical people who might be using a system like that didn't know how to type. They were typing you know one finger at a time and so on. And so it was a big kind of uh uh it was important to have short command names because ideas like autocomplete were far in the future and we were dealing with um dumb terminals where you just typed a character and uh and that would get sent in. It wasn't something where there was feedback to the terminal and so on. Um even there was uh when one was using SMP one could use a teletype, literally a thing that was producing you know pieces of paper coming out of it um and one was was typing characters on that. So it was kind of a different world in terms of what one could do with kind of the names of functions and so on. And uh in SMP I used short names for functions and uh uh that made SMP code very hard to understand. If you saw I can't think um oh gosh what were some examples um uh you know ar for array. Okay what does ar mean? Who knows? um something you know all the various mathematical functions had been named. I think I think those were a little bit longer like BessJ was a Bessel J function. The uh there were functions um uh I think the the the king of of weird function names in SMP was rex uh like the Latin rex uh or the tyrannosaur version. Um Rex was a random expression generator and but if you saw that you know in a piece of code it's like what on earth does this mean? So I had the experience of using, trying to use short kind of shortenings of English words as command names in SMP. It was not very successful there. For APL it was kind of a disaster. Um and really really damaged the potential use of APL. APL has a lot of very interesting design ideas about array programming and so on, many of which I've made use of in Wolfram Language, but APL itself was really never successful on a broad scale. I mean as as happens often there were a small number of finance companies for example which uh sort of exotically chose to use APL for a lot of things but otherwise it was was not very widely used um and uh uh and I think the strange character set didn't help. APL was uh Ken Iverson had worked at IBM and APL was sort of connected with IBM which was at that time the world's by far

largest computer company
and IBM had to develop you know special APL keyboards so people could type APL
and so on. Of course in modern times with soft keyboards and all this kind of thing it's a bit of a different story
but back in the 1960s and early 1970s um that was the only way to solve that
particular kind of problem. It's worth realizing that in those days the standardization of uh kind of um
characters, computer characters, things like ASCII, that was not yet standardized.
IBM was pushing a a thing called EBCDIC, which was kind of an alternative way of encoding characters
that had different special characters. I forget exactly what it had. I don't know whether it had a hash sign, for
example, in EBCDIC. I think it did not. Um so there was a time when when there were sort of different character
sets and the APL character set was such a character set it just didn't happen to make it. So in any case back to the
question of sort of the evolution of language and uh it's it's the evolution
of computational language like Wolfram Language. I mean I guess Wolfram Language is really the only serious example in
modern times of a of a true computational language. There are many examples of programming languages which
try to sort of describe the operations of a computer. What Wolfram Language is trying to do is to describe the way
the world is in computational terms: a somewhat different objective, where we end up right now with about 7500
different built-in functions. Typical programming languages are dealing with a few tens of primitive operations and
then various kinds of libraries that add on to that. So I think the the main
driver of kind of change in Wolfram Language uh well is the the the zero-order
level driver is things we figure out that we can kind of represent in
computational terms, adding on to the language. The other driver is
ambient understanding of things that makes it possible for us to use terms that weren't there before. So you know
there are plenty of things to do with the web. Wolfram Language was first developed before the web was a thing. In
fact uh in a footnote to history uh when when Mathematica, now Wolfram Language,
first came out in 1988 one of the computer systems that it was announced
to be on, although that computer hadn't yet shipped, was the NeXT computer which was Steve Jobs's project between
his time at Apple and his later time at Apple. The NeXT computer was particularly aimed at sort of education,
research, development kinds of areas. And we made a deal for for Mathematica to be bundled on every NeXT computer.
Mathematica came out June 23rd of 1988. The NeXT computer I think came out in
maybe October of 1988. In any case, one of the um one of the places that wanted
to… so the calculation was more people will buy NeXT computers because they had

Mathematica bundled on them. And for us it was like well we get paid for every computer that got shipped whether or not people actually used the Mathematica. Um as it turned out it was a good deal on both sides. And um a lot of places ended up buying NeXT computers I would say almost irrationally because they got Mathematica for free on those computers even though perhaps the price of the whole thing, of the computer, was actually more than it would have cost them to buy a Mac plus a Mathematica software system. In any case, one of the places that did that was CERN, European particle physics uh uh research operation and the theory group at CERN got a bunch of NeXT computers to be able to run Mathematica. The person who was the administrator of that network was a chap called Tim Berners-Lee who uh ended up developing the web on that very network of computers. So in a sense the computers that made the web possible were um were ones that that were got in order to run Mathematica. So so that's the way that I know for sure that the web post-dated Mathematica and post-dated the beginning of Wolfram Language. But clearly there are many operations that we can now talk about in um in language that use the term web that didn't make any sense in 1988. Similarly, there are ones that use the term cloud that didn't make any sense until sometime in the 2000s when the idea of cloud computing came along.

So, you know, as the external language and technological language of the world advances, so we can add new constructs to the language. Um the question of sort of the uh dialects and so on and whether whether the the kind of the big goal that I've had is to have this coherent way of talking about everything in computational terms and not to have to specialize into "oh that's for physicists, this is for mathematicians, that's for social scientists" etc. It may be the case that some operation or some construct is relevant mostly to chemistry, but it should be something that kind of uniformly fits in with all the other kinds of things we do in Wolfram Language.

And that's part of the value, I think, is to make it possible to take something that's, you know, a graph that is a chemical graph and have that be a graph that you can do other graph-theoretic operations on and so on. So that's not a place where we um we would like there to be any sort of separation between those two different areas. Now I would say that everybody who uses Wolfram Language ends up using… just as in human language there are certain words that any given person will use more often than others, so it is with Wolfram Language, and so it is with kinds of different parts of the language that people learn more and use more and less deeply. I would say another question that uh I've been long interested in is how do you speak Wolfram Language? You can write it, type it, read it and it is primarily a written language but there are situations in which it would be nice to be able to just say what you want in Wolfram Language. Actually, I had the experience maybe nearly a decade ago now visiting uh I happened to run into a group of kids in California who had been learning Wolfram Language. They were I think 11-, 12-year-olds, things like that, and they had managed to find ways to speak Wolfram Language. They were a little disappointed that speaking it at their sort of full speed I couldn't really understand what they were saying because I'm just not used to hearing it as a spoken language. But I've been long curious, is there a way of making sort of a version of Wolfram Language that can be spoken? You know, you can certainly say things like, you know, Nest, you know, 1 + # &, x, 7 or something, but it's rather awkward. And the question is uh the well Wolfram Language is a tree-structured language. A good idealization uh you know in natural language, a good idealization of natural language is kind of a grammatical tree where you have uh you know something like noun phrase, verb phrase, noun phrase in English and the noun phrase is then its own subpiece that could be you know adjective adjective noun or something and the verb phrase could be adverb verb and so on. It's again a tree-structure kind of thing. In Wolfram Language it's it's much more straightforwardly and rigorously a tree. In natural language it's kind of roughly a tree. Um but the fact that in natural language we can deal with this kind of tree structure and it feels natural to us. I mean the big red dog that jumped over the moon or something, over the blue moon. Um, you know, for us it's not difficult for us to understand all this sort of sub-clause-ing of “the big red dog,” okay, noun phrase, you know, “quickly jumped,” verb phrase, etc., etc., etc. We don't think particularly, oh, there's a bracket there, there's a parenthesis, etc., etc., etc. Now, the truth is when we use human natural language, we do lose it after a certain number of levels of sub-clause-ing. I mean, if I said um “the big red lazy hungry dot dot dot dot dot dot dot dog,” after maybe four or five levels, it would be kind of hard to say what I was talking about. Um and in Wolfram Language, as you build up these kinds of towers of functionality, you do end up quite often with deeper levels of nesting than we are used to in natural language. But I've long been interested in the question of how can we make kind of a flowing kind of sequential way of saying Wolfram Language using kind of the tricks that one has in uh in standard human language. Different human languages have different ways to kind of define what different parts of a sentence are about. Whether it's a tonal kind of thing, whether it's a thing with case endings, whether it's a thing with additional words and so on. There are ways to do that. And I've long thought that maybe it's by adding, you know, extra words like “open” and “comma.” Well, open and comma are very directly the way it's printed, the way it's written. But it would be nice to have something which sort of flows better that is a speakable version of Wolfram Language. But I consider that as yet an unsolved problem. My guess is that the spoken version would indeed have dialects and would indeed evolve in various ways, um even though the written version is kind of locked down and that's not unlike what's happened in a bunch of human languages. Sometimes in human languages the written form comes after the spoken form but in some cases there's kind of a written form and then the way it's pronounced or the way it's uh it's rendered is is different in different dialects.

Let's see. Jack is asking, "When do you think language first became structured enough to support scientific thinking?" It's a good question. I I mean, I don't think there's been kind of a lack of recorded language, and we only have language in historical times when things were written down. And that's kind of the origin like so many things of sort of earliest languages we have: things like the Babylonian languages, things like Akkadian, Sumerian, things like this. Um, it is my impression that those languages are sort of just fine to express scientific and logical kinds of things. Um, I don't think there's been really, you know, I think we could be speaking Akkadian now and we'd be able to say the kinds of things that uh that we're saying. Now, of course, there's lots of vocabulary that's come in, but I don't think structurally there's a problem with those languages. I think um uh you know I will say that um a good friend of mine actually has the curious job, has had the curious job, uh he is a professor in Oxford and uh of giving sort of, writing Latin orations that can be used for honorary degrees uh given to people in Oxford. And those honorary degrees might be people in computer science or molecular biology. And obviously in in the time when Latin was a natural language so to speak, being a routine spoken language, there weren't words for things like uh you know computers and uh uh and molecules and so on. And so he has sort of the interesting challenge of making up kind of, from what was describable in Latin, sort of renderings of things which we now know in modern times as uh you know chemical bonds or something like this. So it's doable and I don't think there's anything sort of structurally lacking in certainly a language like Latin or even in the earliest languages that we know from uh from written records. Um it's an interesting question how human language evolved and whether uh you know at what point the kind of compositional structure of language, this whole idea of being able to have sort of phrases nested within each other, being able to put together different kinds of things, sort of an adjective

goes with many
different nouns and so on. When did that arise in the history of human language? Nobody knows. Um it's uh those things
when it when it comes to you know animal communication uh that's much more in the
you know “fetch,” “sit,” you know one word at a time kind of setup rather than this
kind of flowing compositional language that we have. Uh I don't think we have any way to know when sort of
compositional language arose in sort of the history of our species and so on. Um I think uh unlike kind of the actual
sort of evolution of language itself where we can kind of see well you know these languages, these Indo-European
languages, have lots of words in common and they're different from the words in you know an isolate language like Korean
or something or Basque or whatever. Um it's you know we can kind of see the tree
of languages by looking at
the commonalities of words that emerge in current languages but all current languages have this
compositional structure and although they do the composition differently — they might have case endings, they might have
extra words and so on — I don't think that's been significant in… I don't think that allows one to go backwards and kind
of look at the phylogeny of language and and see when those things originated. I I just don't think we
know. Um and I think uh uh it's it's probably the case that it's tough to
talk about the kinds of things that one needs to talk about for science or logic or whatever else if you're just in the
one-word-at-a-time phase of language communication. Of course, the the challenge then becomes, and I I wrote
about this recently because I was thinking about what if we go from sort of the cats and dogs with, you know,
a thousand times fewer neurons than we have to our 100 billion neurons to um uh
to uh some entity with trillions of neurons or something. You know, what
kinds of additional things can happen in the structure of something like language? What what comes beyond the
compositional language that we know and what kinds of thinking might be enabled by that sort of higher-level language
that are not things that we currently find familiar? And I think there is some indication of that in fact from what we
built in computational language in Wolfram Language. Um I think there's sort of an indication of a
little bit about what that might look like. Uh it's kind of the notion of well there are there are things that are
just verbs that say “do this action.” But then there are sort of higher-level verbs that kind of describe kind of meta-
actions that you can take that describe sort of how to construct an action. So
even even something like uh oh I don't know, a function, I mentioned the function
Nest earlier. The function Nest takes a function, takes something which does an action like

blurring an image and it takes that action and it then does a higher-level thing of saying nest that action some number of times. And that's not something that's that's very common for us to think about in traditional human language. I think it's something which becomes very natural when it's something that you're going to use in a computational language. And I think that gives one some indication of what it might feel like to have sort of a higher level of language um enabled by kind of uh the sort of bigger brains or or whatever else. I mean, it's a little bit uh a little bit humbling to think, you know, if you're a cat or a dog and you're hearing all this complicated human language and you're hearing kind of a description of how to do algebra in that complicated human language and and but your way of thinking about language is just kind of one word at a time. Um it's one meow at a time or whatever. Um it's it's kind of uh it's a bit humbling to realize you might just completely not get what's going on in compositional language. And at another level, it might be the case that there are sort of higher levels of language that are beyond what we humans have reached that we just completely don't get. Now, it helps when we can use computational language as a way to kind of build a formal tower where we can see how this piece fits onto this piece. We can see this tower built up. We can see how it's applied. That's our kind of crutch for getting to something which might be beyond what we can sort of naturally do just raw with our brains.

Let's see a question here uh from RBS. Was Leibniz's *characteristica universalis* basically a proto-language? Was his main limitation the lack of hardware to execute simulations? So, Leibniz lived in the late 1600s and uh he was um… his main day job was being sort of a courtier and a figure-outer of things. Like he was um uh supposed to kind of write the history of the Dukes of Hanover and figure out, you know, why it was okay for some particular duke to marry some person from some other family, etc., etc., based on the long-distant history of “yes this bloodline was really okay for this or that thing.” So he was a sort of a figure-outer of things and he was also to some extent a bit of a diplomat. I think he was um uh but he spent most of his life kind of hanging around the court of the Duke of Hanover. Um he almost ended up coming to the court of what was it? George III, maybe George… I forget. I'm I'm I'm losing track of my British monarchs there. But um that um that person came out of the Hanoverian uh dynasty and but became the king of England. And in fact, you know, one of Leibniz's uh uh very sort of unfortunate things was his kind of competition with with Isaac Newton. Had Leibniz been brought to the court of George, maybe it was George I, um he would have been sort of higher in rank than Newton was in those days. Newton was by that time controller of the mint, but uh uh you know, Leibniz would have been a direct, you know, courtier-to-the-king type thing. That didn't happen because at the time Leibniz was supposed to be finishing some history of the Dukes of Hanover and he hadn't finished his homework and so he wasn't brought to that court to go to England and uh you know Isaac Newton kind of ran these rather um uh somewhat I would say corrupt kind of um uh investigations into you know the Leibniz-Newton controversy about the invention of calculus, running through the Royal Society which Isaac Newton was was a big figure in and was kind of pulling the strings behind the scenes so to speak to come up with the conclusions that were favorable to him. But in any case, um, Leibniz had sort of a long-term interest in kind of making things symbolic in a sense. His original PhD thesis had been about resolving legal cases using logic. And he was, I think, somewhat uh influenced by a chap called Ramon Llull from the 1200s who had had uh this way of kind of having these uh sort of physically made-up wheels and so on where you could kind of choose different possibilities for what would happen and you could kind of combinatorially enumerate the possible things that would happen. Not not not so different from things like the I Ching and so on, but those were sort of ways to kind of take what happens in the world and have a formal representation of that. I mean, back from Aristotle and his predecessors in the development of logic, there was this idea that kind of the structure of arguments can be represented in a formal way. It's worth remembering that logic as we know it today with ANDs and ORs and NOTs and so on and being able to be used in computers and such like — that was not the logic of Leibniz's time. The logic of Leibniz's time was syllogistic logic where it was patterns of argument like, you know, “All men are mortal. Socrates is a man. Therefore Socrates is mortal.” That's a particular syllogism, a particular pattern of uh of sort of structured presentation of information that sort of makes logical sense. And back in the Middle Ages, people would memorize these different kinds of syllogisms. They had acronyms. There was one called *Barbara*, one called *Celarent*. I don't remember the others that are easy to construct now from the kind of more mathematical structure of logic that we have from Boole from the 1830s and so on. Um but uh in those days logic was memorizing these kinds of structures of arguments. And so for Leibniz it was: could legal cases be resolved by just sort of putting these templates onto them, to say there's sort of a symbolic representation of what happens in these legal cases and that's how we can resolve them? Didn't work out for

Leibniz. We're we're now at a time when computational law is finally starting to be a real thing. Um but that was 300-and-something years later. But in any case in Leibniz's time, Leibniz was also sort of in his diplomatic role. One of the things that was happening at the time was Latin was going out as kind of the uniform language of learning. Up to that time, sort of any serious book written intended for sort of the learned public would be written in Latin. But at that time there were starting to be books written in English, French, German and so on. There was sort of a fragmentation of learned works into those different local languages. I mean notably Isaac Newton when he wrote in 1687, he wrote his *Principia Mathematica — Mathematical Principles of Natural Philosophy* — that was written in Latin. By the time he wrote his *Opticks* it was written in English. So that was right at the end of the 1600s, right when Leibniz was active, it was just the time when there was a sort of breakpoint between the uniform international Latin and the local languages being used. And Leibniz was sort of concerned about that and wanted to kind of uniformize the language of science, the language of sort of structured thinking in some way that was not specific to a particular human language. And so he was very concerned with things like mathematical notation. You know, he invented the notation for uh for integral signs. He invented the notation for derivatives d by dx. He said very immediately, he said people will think you can cancel the d's and that might be a problem, which is of course has been a problem for for calculus teachers for the last 300-and-something years. He also uh argued against William Oughtred's invention of the multiplication sign, saying people will confuse it with an X, which again has has actually happened. But Leibniz was was very concerned about this kind of standardization of notation for mathematics which was just coming in at that time. He also had this idea of making kind of this universal language that would be sort of uh would transcend human language and be a way of representing thought. Now there were other people also thinking about this kind of thing, like a chap called John Wilkins in England was doing this. They were trying to create what they were calling philosophical languages: languages that could represent ideas without being committed to particular human languages. Now in making those philosophical languages, they were inventing their own sort of human-like languages. At the time, there was a lot of confusion between kind of the actual morphology of the language — the way it was written, you know, what kind of glyphs it would use and so on — and its semantic meaning: the kind of separation between semantics, the meaning of things, syntax, and uh even phonology of uh of how it's spoken. The fact that those can all be really factored out and separated, that was a thing for a much later time. So at the time when when

Wilkins for example was
making his philosophical language he had its its own script and and everything else uh made it even more inaccessible,
so to speak, than it might otherwise have been. But in any case, Leibniz kind of had this idea of this universal language
that could represent thought in this kind of almost symbolic way in the way that was kind of… had the structure that
he was imagining would be in mathematical notation, he was imagining would be with these templates for logic
and so on. He never really managed to get that far with that. He um he tried
to convince some Duke of Hanover to create a library that would sort of have
knowledge encoded uh presumably in this kind of universal language. It was kind
of a Wolfram|Alpha play a few hundred years before Wolfram|Alpha. Um
it was uh… it didn't really get off the ground like many of Leibniz's projects didn't really get off the ground. Now
how that relates to other kinds of things that Leibniz talked about. I mean Leibniz did a lot of work in philosophy.
Um and uh I've always found his philosophy quite hard to understand. I mean there's sort of a direct line from
Leibniz to Kant. Um and uh that direct line I think very much leads to ideas
like the ruliad and very much to our modern ideas of space and time and
the role of the observer and so on. But for Leibniz uh there were a lot of things that I've always found very hard
to understand. And Leibniz had this notion of monads — that everything would consist of monads. One of the things
very important for Leibniz was the notion of sort of souls. And where do souls come from? It's worth
understanding that this notion of a soul, it's pretty ancient in theology,
is a notion of something that… there is a thing to brains that isn't just the wet-wear of the brain. There
is some kind of eternal kind of engram of thinking which in modern times we
would think of as computation. But in in earlier times it was this thing very
much tied up with religion that was the soul. This thing sort of that is an
engram of thinking independent of the actual hardware of brains. And as an
engram of thinking, it is inevitably eternal. Just as we might say, you know,
this algorithm we wrote, it's an eternal thing. The fact that it does not persist in the RAM of our computer for very long
is not really relevant. There is an eternal version of it. But Leibniz was very concerned about sort of how the
soul, whatever that might be, emerged in things. And he had this idea. He didn't
realize that sort of soul could arise from non-soul. He didn't realize that
inanimate matter could sort of be aggregated into a brain that
could think and so on. So for him his monads which were elements of everything — they're sort of his ultimate
atoms — were also things that had little fragments of soul in them because he
didn't understand the possibility that one could have soul emerge from non-soul. So that makes

what he talked about with monads really hard to understand because that's kind of not the way we think about things. For example, for the ruliad, we imagine that the ruliad is ultimately made from kind of atoms of being, atoms of existence. Um and that an atom of existence is just an atom of existence. It's a thing without kind of any structure, without anything other than its identity and its distinction from other atoms of existence. That's rather different from Leibniz's monads which were in a sense kind of uh the whole universe packaged into this one atom and then the universe made from many such atoms. So a slightly different kind of idea, but the notion that Leibniz had that it should be possible to sort of make a symbolic representation of everything, that's certainly a notion that that I've tried to continue and and maybe we are we are getting closer to being able to complete the vision of kind of having this computational formal symbolic language that represents the kinds of things that we humans are interested in talking about.

Let's see a question from Graeme here. “Do you think linguistic evolution behaves like computation? Simple rules branching over time into complex forms. How does language evolve? How does language emerge?” Uh I I do think that a bunch of the ideas that we've seen that I've worked on the last year or so about biological evolution and kind of the way of thinking about um biological evolution as a computational process — excuse me — I think those ideas are applicable to linguistic evolution. I mean, the thing that's always interesting about sort of linguistic… what is language? What is it doing? It's taking all those random nerve firings in our brains and finding a way to package up the important aspects of what's going on in there, or at least the aspects we think of as important, in a way that that we can communicate them to another brain. I mean, you know, as I'm talking to you, there are all kinds of random nerve firings going on in my brain. Those nerve firings are particular to the structure of my brain. If I were to pick up those nerve firings and hand them to somebody else and implant them in their brain, they wouldn't work because the brain is set up differently. And so we have to have a sort of transport layer to go from the innards of one brain to the innards of another brain. And that transport layer is primarily language. And so then the question is well what should be in that language? How how should that language be able to recreate sort of a feeling, a thought, from one brain in another brain? What does kind of the transmission channel look like? It's sort of interesting that in neural nets and machine learning and so on, in AI systems, there's this notion of autoencoders where you're

saying we've got
all these different uh kinds of images of cats and dogs and everything else and then we're we're somehow going to uh be
able to uh to get the neural net to sort of compress those images down to some
representation that has sort of fewer bits in it, many fewer bits
than the original images, but then we're going to be able to reconstruct those original images or something like those
original images sort of on the other side. And that notion of a kind of an autoencoder that's kind of squashing
down something to encode a thing to recreate the thing later — that notion of
doing something like that is very similar to what language seems to be doing. And there's a good question of
whether we can kind of recreate kind of the formation of language by seeing sort
of the training of something like an autoencoder. And I think so. I think this question of sort of how, you know,
does language sort of emerge by starting to take somebody's thoughts, communicate them to somebody else — how do
you how do you crush down, compress, package those thoughts so they are communicable? And it's interesting, I
haven't done experiments on this. A number of other people have tried doing these, about sort of how this can
actually work and does language of the kind we see with its compositional structure and so on emerge from
sort of trying to push things into that in that way. I think it's an interesting possibility. I think it's
very likely that something like that can work. Um the things I've done in studying biological evolution,
where there's a certain fitness function — you're trying to get a structure which starts from its underlying rules, runs
those rules, and then succeeds in, let's say, living for a certain time and then dying out — um there you're able to
sort of tweak those rules progressively and adaptively over time to get uh more
and more successful fitness for the organism. So a reasonable question is: if your fitness function is "can these two
critters understand each other when they communicate in this way," what would one expect to see in the formation of
that communication channel? What would be the thing that one would expect in that communication channel? That's a good
question. It's something I've been interested to look at. I have not looked at that. Um I think that the uh the
question of once you have a language and it's being operated in the context of society, how does it then evolve?
I suspect that quite a bit of that is tied up with the evolution of society and a
lot of kind of the structure that we see in the thousands of languages — you know, 5,000, 7,000 languages or something in
the world. Um, a lot of the kind of details of those languages are determined by a lot of kind of uh

demographic, sort of geographical information. Whether it's “oh there's a mountain range here that separates these two different communities that have never had to talk to each other” or “oh this community came in contact with this other one and that formed some kind of creole language” or something. And it's you know, there's a lot of sort of human geography that's affected the formation of languages. And I think it's a question to what extent there are sort of universal principles of the way that languages evolve. I mean the Brothers Grimm in addition to collecting fairy tales also were trying to collect information about the evolution of languages in Europe um and that that leads to things like Grimm's law which is a law for the way that pronunciations evolve over time um and so on. And I think it's a reasonable question: is there sort of a universal evolution of language that's inevitable or is it all determined by kind of details of society and human geography and so on and and and for that matter the creation of technology? Because the new words that arise are often words where they either come from new sort of social structures or they come from new technology, new discoveries and so on. I mean if we look at English for example uh you know there are certain constant progressions that I've certainly noticed. Uh one of them is the smooshing of words. Back in the day it was “web space site.”
A few years later it… “website” is a word. Um and that's a a very common thing. I may have always pointed out to our document quality assurance team that uh looks at documents, among them things that I write, and says “that word is spelled wrong” or whatever else it is. Um you know I always tell them, you know, when it's something which is a two-word thing for example it's like over time those words will smoosh and we might as well be ahead of the curve and put “website” as one word rather than two.
But that's an example of something which seems to be a progression that might be sort of a natural progression. As soon as a word is used a lot, things tend to smoosh together.
Of course, that's not always the case. I mean, in in uh I don't know, in French there's “s'il vous plaît,” for example, which remains as a um as a what, almost four-word combination, uh for you know the word for “please.” That hasn't smooshed into a single word. So although it's always said together. But I think in um uh… but this question of sort of the evolution of language: I don't know to what extent it can be uh independent of the ambient evolution of society and of the kinds of things people want to talk about. Those are all things… the evolution of society is actually something that I'm very curious to study in sort of computational terms and uh to sort of

understand. Uh there are various sort of progressions… actually one thing I'll mention about language: language
doesn't work if only one person speaks it. Language is a is a story of kind of a community where everybody has to speak
the language or the language isn't useful to anybody. And it's kind of this whole notion of kind of the
uniformization of things. Uh something where you might say, you know, in a in a
typical model in physics, it might be there's a gas and there's molecules. Every molecule is for itself bouncing
around. But in fact we can describe the gas just in terms of its overall pressure and volume or something um
for many purposes. So similarly there might be things where uh if we look at
the operation of society for example there are things where everybody's doing their own thing but in the aggregate
there are things you can say about society. And in fact the very possibility of having any kind of narrative of
history. You know, different people may disagree about what the narrative of history should be, but that there is a
collective narrative of history that is even roughly sort of collective among
millions of people is an interesting fact and it's not unlike the fact that we can describe the important features
of the gas in aggregate terms without having to describe what every molecule is doing on its own. And I think there
is sort of a common thread between kind of the aggregate description of
society and the possibility and utility of language as kind of this standardized way of communicating thoughts, but I
haven't unraveled how that all works.

Let's see. RBS is asking um uh was there
a lost century of logic between Babbage and Shannon? Uh, did railway or
hydraulic engineers build Boolean gates without realizing it? That's an interesting question. So, okay, let let's um talk about several different things. So, so Boole was a
contemporary of Babbage in the 1830s, 1840s, that kind of time frame. Uh, Boole really
took what had been this template-based syllogistic logic and said you can mathematicize it. Boole was, you know, knew
about algebra. He knew people like De Morgan, Augustus De Morgan. Um, and uh
he he was kind of like, well, can we shoehorn kind of the operations of
logic — things like "this and that," "this or that" — can we shoehorn these
things into something like algebra? And there's a bit of a problem. You could say, well, yes, true is one, false is
zero. Okay, great. AND is multiplication. You know, it's it's false unless both are one and one, then it's always false. The real problem comes about with OR, because
it's like, well, it's 1 or 0. That seems like 1 + 0. Great. That's equal to one. But 1 + 1 is not equal to one in
ordinary arithmetic. So, Boole kind of had to contort himself to make a Boolean algebra in which 1 plus 1 equals 1. Um

and and he did that and uh things started to develop a little bit from that. Um there were various sort of mechanical… well, people like Lewis Carroll Dodgson was um was big in sort of promoting symbolic logic and wrote textbooks of symbolic logic. This was kind of a follow-on to Boole-type ideas. Um and then there were little machines that were built. Uh was Jevons — Jevons, I think — was one person who built such a machine. There's another well-known person whose name I now forget who also had built machines in the 1800s that were kind of logic machines trying to implement Boole's ideas. So those were known things. They were kind of novelty items. They were not part of engineering practice. Now the truth is that I would say in different countries there was a different level of formalization of engineering practice. In a country like France there was quite a lot of commonality. You know there were mathematicians who were being engineers. I would say that was probably less true in England for example. It was much more people like James Watt uh who among other things invented ideas like control theory, ideas like governors and so on. Uh that was not… I don't think he thought of himself in any way as a mathematician um even though he was doing kind of mathematical kinds of things. Um so I I think uh that was um that was kind of a bit more of a divide between engineering practice and kind of theory. Now, Babbage… Babbage was trying to make a computer that would print mathematical tables, that would print tables of logarithms and things like this or more specifically would print kind of values of polynomials that were approximating different things you might want to approximate for navigation, for life insurance, whatever else. Um the uh uh the thing that um um uh… so he never really got into the idea of of building logic into his machine. Uh Ada Lovelace, who I think understood sort of the objective of the analytical engine, which was originally kind of a "we will print different kinds of tables of polynomials," um I don't know whether she understood kind of the fact that there was sort of a logic that could be applied. I mean she was talking about things like, you know, famously she has a sentence, "The Analytical Engine will weave algebraical patterns as the Jacquard loom weaves patterns of pictures of birds and flowers." So she was thinking about it as algebraical patterns, I think. I mean she talked about applying it to the composition of music. She talked about applying it to the computation of things like the three-body problem in celestial mechanics, solving the equations for that. But I don't think it was a story of logic as such. Now what happened in um uh… by the time uh… so in the early part of the uh 1900s there was sort of the rise of electromechanical devices. I mean electrical devices — people like Faraday had been around in the mid-1800s and lots of sort of electrical kinds

of things were developing, people like Wheatstone, later on, well, Maxwell and Hertz by the by the end of the 1800s. But the idea that sort of there would be electrical machines — well obviously Edison and transformers and electrical supply and so on, this was very much a thing by the beginning of the 20th century. And people were certainly connecting all kinds of circuits and knowing that, you know, you could have switching this to that and so on. Uh that was certainly part of electrical practice. The idea of formalizing that and the idea of sort of switching networks that were formalized… well, it didn't sort of mathematicize until Claude Shannon's master's thesis, I think in the 1930s. Um now it will have been the case that there were increasingly sophisticated electromechanical switches used particularly for the phone system um that were coming into play, and and I think, and also to some extent for calculators, but more so for the phone system, and there must have been knowledge, you know, there were crossbar switches and so on. There must have been some knowledge of the fact that that was sort of a logic-like thing. Um I would say that the um uh uh… it was um… well by the time of um uh… by the time of the Second World War and the construction of things like the code-breaking stuff at Bletchley Park in in England, um that was all kind of uh switching, electronic switching uh systems that had come from the phone system. Again, I don't think they even thought about it, even at that time, as a story of logic. They thought about it as a story of kind of uh filling in different features of some kind of possibilities, combinatorial possibilities, for breaking a crypto system. I don't think it was like "this is general logic." In fact, at the time of early electronic computers again it wasn't really a story of of logic as such. The idea, you know, when John von Neumann was writing about, you know, "how should you build computers," uh, he was talking about different kinds of organs, and those organs in a biological sense, those organs were logical organs, things like AND and OR and so on. And I'm kind of thinking that that was how kind of the notion of logic like that came into the construction of electronic computers, because I think in things like the ENIAC and so on the real focus was on doing arithmetic operations and maybe doing some conditional branches and things like this, but that was that was not really the "this and that" and so on.
But that was certainly something that was in, you know, von Neumann's work in the 19- uh 40s and and um beginning of the 1950s on computers.

Let's see. Susan asks, "If you could redesign human language from scratch, what would you change?" Well, that's been tried a bunch. In the late 1800s, uh, things

like Esperanto were kind of brought in as more regular and more international
versions of human language. It didn't catch on. I mean, I remember when I was a kid, particularly at like
Heathrow Airport in London, there would be signs in, you know, English and French and German and whatever and Esperanto.
And it was a bit sad when some new terminal was built or something and the Esperanto signs went away. That was
probably maybe in the '70s the Esperanto signs disappeared. Um, Esperanto never
really caught on. Why was that? Well, really it's because the tide of English
kind of took over and English became the de facto international language and there wasn't really a need for something
different. Um, and I think, you know, Esperanto was all tied up — I think the word “Esperanto” in Esperanto means “I
hope” — um and it was very much tied up with sort of an internationalist movement and a kind of a uh uh a world
peace movement and so on which uh uh probably didn't survive the um among
other things things like the First World War and so on. Um I think that the uh um
so you know one might hope that having a language where kind of it's everything
is phonetic, you can spell easily, that might be a nice thing to have. It's kind of, you know, English has these
weird design hacks where people tried to burn in kind of the meaning of words
into their spelling in very weird ways; like a famous example is “debt,” d-e-b-t,
which comes… where the b is not pronounced and I think in earlier English it had been spelled d-e-t-t, but
somebody had the bright idea that, you know, debit and so on from Latin — let's burn that meaning into the
spelling but not pronounce it — which is sort of a design mistake in English that um
one could expect to, you know, one might not want to have. And obviously languages where the written form
came after… where the spoken… where the written form came after the spoken form,
you don't have things like that. I think in Russian for example it's mostly phonetic
in spelling, which makes, you know, the spelling-bee competition less interesting so to speak. Um but uh uh you
know I think there are there are features like that in English. Um I think uh uh different languages have different…
as I say, different ways to do things. There have been a whole rash of conlangs,
constructed languages, that have been made, I think more often as art-project
objects than for practical purposes. I mean there are, whether it's Klingon or Dothraki or one of the other
languages that's been made for some, you know, movie or or television show or
something like this. Those are constructed languages which are in a sense artistic languages. There are also
languages which have been made again somewhat as art projects — languages like Ithkuil which is sort of the most
complicated constructed language that tries to sort of put all the functionality of all languages into one

language, or things like Toki Pona which is a very minimal language which, if it didn't have kind of the uh the kind of morphological structure of something like Finnish, would probably sound like baby talk. Um but you know that's a kind of a minimal language. I think this idea of, you know, can we make a constructed language that's better than languages we have… I know people have imagined that, you know, learn to speak Ithkuil which has everything built into it and sort of the Sapir-Whorf hypothesis will take you to better thinking. I don't think that's convincingly worked out. Um but that's the kind of thing one can imagine. Now you know I think it is a thing that's worth understanding: in human languages there seems to be pretty good translatability of most things between most languages. I mean there are languages that are spoken, used in places where people just don't have a particular kind of thing. Um, you know, they don't have snow on the equator or or whatever else. Well, maybe they do on a mountain, but um uh you know, so there's more and less need for different concepts in different languages. So there are things which are a bit hard to translate because those concepts don't exist. And there are kind of nuances, cultural nuances, which don't really come with the language, they come with the culture which has been expressed in a language. And those are sometimes hard to translate. But fundamentally there's decent translatability between human languages. And uh uh there's sort of a… so they're all sort of equivalent at that level. And if you can think in in one language, you can think in another language probably with some nuances, mostly cultural, between them. In in computer languages it's actually a very different story I think. I mean, having spent a lot of my life designing and building the Wolfram Language, computational language, I I can plainly see that there are concepts that even for myself, once I've kind of crystallized them in our language, then I can think in terms of those concepts and sort of build towers with those concepts. But without having invented those things in the language I'm sort of just not in a position to think in those terms. It's, you know, one can imagine that there is a way of forming that thought as some kind of complicated incoherent collection of nerve firings. But to make that thought coherent, to package it up so I can remember it, so I can build a tower based on that thought, so I can communicate it to other people, that seems to be a story of having to have some actual language-word-type thing that comes from that. But I think it's something that people uh… in the construction of languages, I mean in in human languages there is uh… people have have long understood that there's sort of all kinds of political communication, that "oh you must say this rather than that." That's kind of a form of kind of changing the way the culture, the way society, the sort of politics of what's happening works. I mean famously in *1984*, what is it, Newspeak or something is kind of a language that is intended to form the way people think about things, and it's certainly been attempted. I think it's been rather unsuccessful mostly in um uh kind of, you know, "you should say this rather than that." I mean, I suppose there's there's a small amount of that in in sort of modern American usage that has has happened, but it feels like the more dramatic versions of that have usually not in the end worked out. But when you're building computational language, it's uh… because it evolves in a way that's really designed rather than something which sort of naturally evolves through usage, and because it is something where the number one objective is you can communicate with your computer, not "you can communicate with your fellow person," um it's… although that is a secondary objective for us, it's something where you can just uh sort of from scratch say "this is how the computational language is going to be set up," and if you do a good job in that um you have a way to sort of scaffold people's thinking about lots of kinds of things and it works really well. Now I have to say, in my life I I've really done this more or less twice: once for SMP, once for for Mathematica and Wolfram Language. And I made a bunch of mistakes in SMP. I had a bunch of pretty decent ideas I think. Um but it certainly was was interesting to see, having designed and built SMP, seeing people use it. It was interesting to see what happened in actual usage. In actual usage people uh… you know, one one example was: I had these things called marks in SMP and I remember kind of the main training documents about SMP had this sentence that said, you know, "marks are the enigma of SMP," which is to a language designer… it's just, you made a mistake in designing the language. People didn't understand it. I would say in language design it's at some level sort of a strangely thankless task, because if you get the design right it just feels natural to people. If you get it wrong people keep on, you know, getting confused and asking questions about it and so on. And what is it… how does one get language right? It's kind of a rather difficult, almost artistic activity, or architectural activity. You know, what makes language feel natural to people? I think I've over the last 45 years probably developed a certain amount of intuition about that and I certainly can tell, "oh, we're thinking about putting some construct into Wolfram Language, it's going to be awkward. People are going to get hung up on that," and "no, there's another thing that's just going to feel smooth. It's going to feel like it fits into everything else that's going on.” Some of that fitting in comes from the ambient use of natural language. Some of it, I suppose, comes from some kind of continuation of a sort of coherent strand of thinking that one develops in building up a computational language.

Let's see. Okay, a number of other questions about language. Let me try and um uh go through some of these and then we have to wrap up here. Peter asks, "Which ancient civilization came closest to developing something like computational thinking?" Um I would say the Babylonians. The Babylonians were systematizers. I mean they had the forcing function of creating the first cities and lots of things sort of had to get standardized and systematized. When you have people sort of living together in a city and things get bigger and you have to have kind of standardized, you know, weights and measures and things like that, it can't just be, you know, “I know my friend Fred picks up, you know, this… he's, you know, giving me three hands’ worth of barley or something and that's this.” It has to be sort of standardized if you're doing it across this kind of larger domain of a larger city-state kind of thing. And I think in terms of the way that um uh you know Babylonians certainly had algorithms effectively for describing kind of astronomical… doing sort of astronomical calculations, things like that. They really had those kinds of ideas. I would say more so than the Greeks actually. Um Greeks were much more “this is the principle of how it works,” I would say much less on the step-by-step “this is how to do it” type thing. Um and then when it uh uh you know when it comes to Roman times and so on and I think there was a lot of kind of “this is how to do it” when it came to military processes, when it came to various kinds of construction processes and things, but I don't think that was so much in the abstract domain. Um, so I would say the Babylonians are are probably the ones. Now, you know, they were lucky because they wrote their kind of… they wrote things down on clay tablets which have survived. I mean another civilization that… where a lot has survived is the Egyptian civilization. It's my impression that there was a lot less kind of of this sort of “let's think sort of algorithmically about things.” I mean, we don't have uh, you know, the Egyptians clearly had design — we even know some of who did it as individuals, the design of the pyramids and things like this — but I don't think we have as much of the kind of “this is the abstract algorithm by which you do this.” Um, we have a bit more of that in sort of the Babylonian way of thinking about it.

And when you say “abstract algorithm,” it's it's not quite as sort of as
procedural as one might imagine that to be, just like the very earliest grammars
were really just aphorisms: “it works this way and this way and this way and this way,” a bit like
the templates for logic and so on.

Let's see uh… I see asks, “Is there any truth to the
saying that the American accent is the original British accent?” You know, this is so steeped in
uh kind of uh uh national
pride and sociopolitical kinds of things. I know when I lived in England uh people
felt, were still feeling, very superior, even the British, you know, by by the
1960s and so on, you know, the US was far the ascendant kind of country
relative to England. I mean, that had been… that was a story from uh… was at least 40 years old
by that time that the
US had sort of surpassed Britain as as a significant power in the world.
I think that the um um… the thing that um people used to say — with complete seriousness —
was that, oh, you
know, the American accent comes from the kind of the farmer's accent in
Cornwall. Cornwall is kind of the… I suppose has the role of something like Appalachia in in US
culture or something, and it's kind of a
you know, it's the hick farmers in Cornwall, that's that's where the the American accent came
from. I I suspect
it's complete nonsense but it was a it was a sort of national-pride kind of
claim that was made.

Um there's a question from Jamie here. “Did
scientific knowledge spread more through people or through written systems in ancient
civilizations?”
Um I would say that that it was a very people-to-people kind of thing. Things
did get written down. I mean, for example, in um uh the Library of Alexandria uh
things were clearly getting written down but still, you know, Euclid was teaching in the Library
of Alexandria, which is
basically a university. Um the fact that his works got written down… I don't know
what the kind of the practice of what was happening, whether it was like “you read the scroll first
and then you go to
talk to Euclid” um or whether it was “Euclid is giving lectures and somebody wrote
down something from the lectures so they could remember it.” A practice um that uh
in in the US at least, and I think in other countries, there were these things called ciphering books
which were things
where when you learned math in school what you did was you wrote down — you
wrote out — this book of different kinds of math problems: the rule of threes, the rule of sixes
and so on, different
kinds of structures of, you know, this… which were all sort of templates for doing math. And
people
would spend, you know, years of their schooling writing out these ciphering books and then they
would keep them as
kind of textbooks. If you were going to be a farmer and you needed to figure out how you would

do some kind of calculation with, I don't know, compound interest or or some land-surveying calculation, you would pick out your ciphering book that you had written when you were a 10-year-old, 11-year-old, something like this. So that was that was kind of the um uh the practice. Um and I think uh it could be that the written works we have from Euclid were somebody writing things out in that kind of way to remember for themselves. It is my strong impression that whether it's Plato's Academy or the Library of Alexandria that it was like humans were teaching humans rather than sort of offline textbook reading. Uh I I don't know whether that's known for sure, but um uh that's a um… that would be my my guess.

Um, let's see. Well, I think I should probably go to my day job. Somebody is commenting that it is Ada Lovelace's 210th birthday today. Um her original birth was a um um uh kind of a a curious and almost scandalous thing. Um her father um uh Lord Byron, George… oh my gosh, what was his last name before he was Byron? Oh, I don't remember. Um but anyway he became Lord Byron, was a celebrity poet um and a celebrity who uh uh that was her father. Her mother was a person called Annabella Milbanke who was a rather straight-laced kind of later activist-reformer type who um uh quite how she had gotten together with with Lord Byron was a was a strange thing and things fell apart before Ada was born. Um uh you know Byron was quite the playboy. Annabella Milbanke was quite the straight-laced person. Um, by the time uh Ada was born, Byron had left England. Uh, he never he never saw Ada in person. Um, he had apparently a locket with a picture of Ada with him, but he never saw her in person, never met her. Um and um the uh uh I think the main obsession of Ada's mother was to prevent Ada from going the way of becoming a crazy poet. And so that was part of why she pushed for Ada to learn all these scientific kinds of things. I mean they lived in a country house and as was the tradition of the time, Ada was kind of homeschooled and um uh she was always getting kind of um uh sort of tutorage in in areas like mathematics and so on to avoid Ada becoming a poet of any kind, even though later on Ada would describe herself as a “poetess of science” um and uh and Ada's sort of uh uh… her contributions were particularly in sort of condensing ideas in the way that sort of a poet-philosopher might do. Um so that was a a um uh that was how that went. But I think um the uh the turbulence of Ada's birth uh sort of reverberated for for well over a century as um uh the kind of supporters of of Annabella Milbanke and supporters of of Byron kind of duked it out with with books written by each side and and so on uh talking about the shocking features of um of each side. And I think Ada got somewhat caught up in that. And there was lots of lots

of scandalous
things said about the later Ada as a result of that kind of um… the supporters of Byron were were always trying to sort
of character-assassinate Ada as a way of kind of um character-assassinating her mother. I mean her mother outlived Ada.
Ada died quite young of probably cervical cancer. Um and um the uh her
mother was was always, I think, um… I think her mother was a somewhat difficult
character. I mean she was always explaining that, you know, she was very sick and so on even though she lived to a ripe old age um and uh was always kind
of using that as a as an excuse for different kinds of things. Although the one thing that perhaps Ada's mother did
contribute to the course of intellectual history was that um the uh
she took Ada as part of her kind of reformed-society thing — this was a time
of the Industrial Revolution and uh the the… well, what were later described as
the dark satanic mills of northern England — um she uh she took um Ada on
tour of the mills um in which, when Ada was probably 15 years old maybe, and uh
Ada had the chance to see all this machinery. And that was the same kind of machinery and the same kind of ideas
that uh ended up being used by Babbage in his uh his Difference Engine and later
Analytical Engine. Ada had seen that kind of machinery uh by being toured around the mills — I think more for sort
of reform purposes than other things. I mean, Ada's mother had started a school, for example. Um and uh and various other
things. Anyway, much to say about Ada. Um always an interesting topic. Um I
think uh uh she would have been interesting to meet. Um I think a a
mixture of a British-society person with a math nerd. Um and uh uh quite um uh um
a spirited person, I think. Anyway, I should wrap up there. I think I have a
day-job livestream coming up right now, but uh thanks for joining me and uh talk
to you another time.