The Chaitin Interview IV: Knowability and Unknowability

https://mindmatters.ai/podcast/ep127

Robert J. Marks:

Can we prove things exist that are unknowable? That's the topic today on Mind Matters News.

Announcer:

Welcome to Mind Matters News where artificial and natural intelligence meet head-on. Here's your host, Robert J. Marks.

Robert J. Marks:

Greetings. We are talking to Professor Gregory Chaitin, the co-founder of the field of algorithmic information theory that explores properties of computer programs via their length and their degree of compression. Professor Chaitin, welcome.

Gregory Chaitin:

Glad to be with you today.

Robert J. Marks:

Okay. You wrote a book for Springer in 1999. It was called *The Unknowable*. And also there was a tribute book to you, *Unraveling Complexity: The Life and Work of Gregory Chaitin*, and you were solicited to be an author of one of the chapters. I think it was the second chapter, and it was entitled "Unknowability in Mathematics, Biology and Physics." This is a big deal. What is unknowability from your perspective?

Gregory Chaitin:

Well, okay. So I have a number I was proud of discovering called the halting probability omega, and that is something where you can define it mathematically very simply… well, maybe not that simply. The bits, the base two bits of the numerical value of the halting probability omega are maximally unstructured and maximally unknowable, it's irreducible mathematical information. That's actually not so easy to prove, I guess you have to go through a course on algorithmic information theory to do that. But the ideas are simple… that you can't prove that programs are elegant is a much simpler incompleteness result.

Gregory Chaitin:

Now, the question is: Does that number omega exist? It exists in the mind of God. To an infinite mind, it is knowable, you could calculate each bit, but we're not infinite minds, right? You can argue that it doesn't even exist, that it's a fantasy object. Mathematically, it doesn't use very sophisticated math. There's much wilder mathematics. But the numerical value of this number omega is maximally unstructured and maximally unknowable. So compared to the kinds of things you do normally in mathematics, this is a forever unknowable thing… the value of each individual bit in the numerical value of this number.

Gregory Chaitin:

But you can counter-attack saying, "This is a fantasy." The real number to an engineer is a fantasy. The real number has infinite precision. Engineers don't work with infinite precision. We're happy to get a few decimal digits. Different problems need more precision. But infinite precision is nowhere to be found except in the imagination of mathematicians, who talk about a real number, which is determining the position of a point with infinite precision. But this fantasy makes the mathematics simpler. It makes for beautiful mathematics, and a lot of physics is based on partial differential equations and depends on the fantasy of infinite precision, real numbers. So, in a sense, real numbers are justified by their practical applications. But nowadays, not so much, because people don’t solve equations that much anymore to do engineering. They use computer simulations.

Robert J. Marks:

Or they go to Mathematica.

Gregory Chaitin:

Right, which is using finite precision mathematics, not infinite precision real numbers like you do when you have equations, and you want to prove properties of the solutions of the equations. This is how things used to work, right? Or you would find an analytic solution to an equation. That only works for very simple systems that you can solve in a closed form mathematically, and that’s disappearing into the past. People are more concerned now with calculating than they are with proving theorems. In that sense, I’m a dinosaur.

Robert J. Marks:

Well, in fact, we have a name for them in engineering we call them keyboard engineers. If they're presented with a problem, they don't go to the theory which gives you depth and insight into what's happening. They go to a keyboard and try to work things out just to get a surface sort of answer.

Gregory Chaitin:

And that's good enough for most practical purposes, right?

Robert J. Marks:

Yes.

Gregory Chaitin:

Instead of messing around with very complicated equations and trying to find an analytic closed-form solution, you simulate the system and see how it behaves, and you tweak the design a little bit to see if it behaves better, more like what you want. So that’s a new paradigm that the computer enables us to use.

Gregory Chaitin:

So you can argue that the halting probability omega number doesn’t exist. Furthermore the halting problem contemplates an infinite amount of time for the calculation. Will the program halt if given an arbitrarily large amount of time? But computers will not last for an arbitrary amount of time. They’ll break or the earth will freeze or the sun will go nova. The halting probability is like talking about unicorns or flying horses.

Robert J. Marks:

Let me tell you a pushback that I got from the idea of unknowable. I mentioned I'm a big fan of your proof that elegant numbers are unknowable. I think it's very, very insightful. It's beautiful.

Gregory Chaitin:

It has a very simple proof.

Robert J. Marks:
It's a simple proof, and, I think, according to Paul Erdős, it would probably go into God's Book as the simplest explanation of something.

Gregory Chaitin:

Thank you. That’s the nicest thing anyone ever said to me. Most people prefer the omega number. It fascinates people. A religious person in Vienna once told me that he viewed it as a step closer to God, because the numerical value of omega is in the mind of God, but we can’t find out what it is, right?

Robert J. Marks:

Yes. Well, I think also that your omega, your work on the omega number, as it's referred to as Chaitin's number, it also belongs in God's Book, so it's also beautiful.

Gregory Chaitin:

Oh, thank you. Jack Schwartz, though, he never liked my work on the omega number. He liked the proof that you can't prove that a program is elegant.

Robert J. Marks:

Oh, really?

Gregory Chaitin:

Yeah. And I greatly admired him.

Robert J. Marks:

And we're going to, in a subsequent podcast, get deeper into Chaitin's constant or what you call omega's number. Here's the pushback that I got back from this, Greg, is that I mentioned this proof of your proof of elegance, that elegance was unknowable to somebody, and they say, "Well, that doesn't mean it's unknowable. It means only that it's non-computable." And I guess my response was, "Yeah. Non-computable, does that equate to unknowable?" I think that that's the assumption here. And what's your take on that?

Gregory Chaitin:

Well, there’s no way to calculate if a program is elegant. And there’s no way to prove it in individual cases. So that sounds like unknowability to me. Let me talk about the halting problem, which in theory is undecidable. But if you limit the amount of time to some reasonable amount like one day of calculation, which is certainly a lot, you just run a program for a day and you see whether it halts or not, and if it hasn’t halted, you just assume it’s never going to halt. For all practical purposes, that may be a good answer. If there is no time limit then you are in big trouble, but that’s a mathematical fantasy, you see.

But mathematics deals with fantasies, because they have clean, beautiful properties, and you can prove theorems about them. So mathematical fantasies can be useful, for example, in theoretical physics.

Gregory Chaitin:

Let me give another example: If you toss a coin a lot of times, almost certainly the result is algorithmically unstructured or random, it cannot be compressed into a smaller program. And you can even get estimates of how very probable this is, and it can have enormously high probability. So toss a coin *n* times and get a lot of zeros and ones. Most probably the resulting sequence will be very close to maximum program size complexity. There won't be any program for it substantially shorter than *n* and you can quantify that, what the cutoff is. So something is almost certainly true, but you can never prove it from mathematical axioms that are less complicated than the number of bits of the sequence you are trying to show is random, is unstructured.

Gregory Chaitin:

And there's also the question of which axioms you are using. My belief is that pure mathematics should evolve, should be creative, that you can add new principles. And, in fact, during my lifetime, I've seen a number of new principles added to pure mathematics.

Gregory Chaitin:

So these questions are all complicated to discuss. Like most philosophical questions, on the one hand, this and on the other hand, that. And they're all good arguments, and you pays your money and you takes your choice, right?

Robert J. Marks:

Yes. Yes. So elegant programs, just to remind the listeners, an elegant program is the smallest program to achieve some objective. There exists a smallest program. So that is the elegant program. You can also think of it as a... I'd like to think of it in terms of images. If you have a big image, what is the most that you can compress that image, and that would be the elegant program for the image.

Robert J. Marks:

Greg, I know that you've spent a lot of time with your proof that elegant programs are unknowable. Do you have a simple explanation that you could walk through that people of modest education could understand?

Gregory Chaitin:

Well, I try.

Robert J. Marks:

Okay. Let's give it a try.

Gregory Chaitin:

I try to explain the proof in my "Experiment in Autobiography" that's included in the book, *Unraveling Complexity*… By the way, there's another festschrift that they did for me when I was 60. *Unraveling Complexity* was intended to be for my 70th birthday, but it dragged on a number of years… I have an autobiographical essay in *Unraveling Complexity* and in it I go through what I see as the essential ideas in the proof. I've often tried to do this when I'm invited to give a talk at a university, and there's a funny test for whether I explained it well. When I explain it well, people laugh. When they realize the paradox of how the proof works, people laugh. And when people stare at me with glazed eyes and just don't react, that meanst I didn't explain it well. So let me see. I can try to get a even simpler explanation than is in the "Experiment in Autobiography."

Robert J. Marks:

Well, Greg, now I'm paranoid. I don't know. I hope I laugh after your explanation. We'll see.

Gregory Chaitin:

Well, okay. So let's discuss provably elegant programs. Provable means provable from a fixed set of axioms and rules of inference, right? That's a formalization of the axioms for mathematics. And Hilbert thought there was a formal axiomatic theory that all mathematicians could agree on. And once you've taken all the subjective element out of it, it's like a computer program. Then you just run through all possible proofs, you check which ones are correct, you filter out the correct proofs, and that way you get all the theorems. All the theorems of mathematics it would be, if Hilbert had come up with a formal axiomatic theory for all of math.

Gregory Chaitin:

Okay. So now whatever the system you're looking at with whatever the axioms and rules of logic, it can be implemented as a computer program. It's sort of the proof-checking algorithm, which always gives an answer… the proof is correct, the proof's incorrect… and from this to an infinite-runtime algorithm that generates all the theorems in order of the size of their proofs is but a small step. You look at the size in bits of the proof-checking algorithm or of the algorithm that runs through, checks all possible proofs and gives you all the theorems. You look at the size in bits of this. This gives you the algorithmic information content or the complexity of that formal axiomatic system that you're studying to see what it can achieve.

Robert J. Marks:

If I remember right, you step through numbers one at a time and check if it's meaningful or not, right?

Gregory Chaitin:

You check through all possible proofs one at a time.

Robert J. Marks:

And those correspond to numbers. Is that correct?

Gregory Chaitin:

Well, they correspond to strings of characters in the alphabet. You can also just go through the tree of all possible proofs. That's another way to get all possible theorems one by one… an endless computation. The program for this will require a certain number of bits, that's sort of the bits of axioms you're using in this formal version of mathematics, which Hilbert thought would be all of mathematics.

Gregory Chaitin:

So let's say the program that does this, either the proof checker or the one that went through the tree of all possible proofs or the one that checks all possible proofs in size order and gives you all theorems… Let's say this is *n* bits, whatever that is. So now you start running through all possible theorems until you find a proof that a program that is substantially more than *n* bits long is elegant, which means it's the smallest program that calculates the output it does.

Gregory Chaitin:

Okay. So you want to see if your formalization of mathematics and all its principles enables you to find a provably elegant program that is larger than the program which is the software embodiment of those mathematical principles, of that formal axiomatic system, its axioms and its logic, rules of inference. You keep running through all the theorems, all possible proofs and all the theorems, until you find a provably elegant program that is substantially bigger in size than the software implementation of this process to run through all possible proofs and get all the theorems.

Gregory Chaitin:

And then what you do is you take that provably elegant program and run it, and then you see what its output is and this is your output. So what we have is a process, a program, that is basically the number of bits in size of the number of bits of axioms in your mathematical theory. And we're using this theory to attempt to prove that programs are elegant, and we keep looking through all the theorems until we find a proof that a string of zeros and ones, a finite string of zeros and ones, that is substantially larger than the software implementation of your mathematical system, is elegant.

Gregory Chaitin:

So then you run this program, and its output is your output. Okay? This is a process you do that will come to an end and give you an output. Now look at this program that I've described in words. It has a number of bits which is basically the number of bits for the software implementation of your axiomatic theory, and it's giving you an output which is the output of the first provably elegant program that is substantially larger than the number of bits in your mathematical theory. Now, this is a contradiction, because you've proven the program was elegant from these axioms, and that means that this is the smallest, most concise program that can calculate the output that it does. But you've calculated the same output with a substantially smaller number of bits, because you found it by running through all possible theorems, all possible proofs, in your mathematical theory. And by the construction of this paradox, you keep running this process until you prove that a program is elegant that has substantially more bits than the program which is the software implementation of your mathematical theory.

Gregory Chaitin:

So now, this has given you a smaller program than the supposedly elegant program to calculate the same object, and that's impossible by the definition of elegance. So either you're proving false theorems or you can never find this program. If you can only prove true theorems, the elegant program for the thing you calculated is substantially larger than the program that found this provably elegant program. So you've actually compressed the output from this supposedly elegant program even more into a shorter program, and that's impossible by the definition of elegance. So if you assume that only elegant programs can be proven to be elegant… in other words, that the theorems you're proving are true… you can never find a proof that a program is elegant if it has substantially more bits than the software implementation of your mathematical theory. The only way you avoid the contradiction is if this process never finds a proof that this program that is substantially larger than your software implementation of your mathematical theory is elegant.

Gregory Chaitin:

Okay. So in other words, any mathematical theory can only prove that finitely many programs are elegant, they have to be smaller in size than the software implementation of the mathematical theory. But there are an infinite number of elegant programs, because -

Robert J. Marks:

They get bigger and bigger and bigger.

Gregory Chaitin:

Because for any programming task, there is a most concise program for it. The problem is you're not going to be able to prove that you've got the smallest program if the number of bits in your program is larger than the number of bits in your mathematical theory. I don't know if this was understandable. It depends on the notion of a completely formalized objective, not subjective, axiomatization of mathematics. That's what has limitations, not mathematicians. Gödel thinks that mathematicians are not limited by his incompleteness theorem, that in fact they can directly intuit facts from the Platonic world of ideas. So this incompleteness result that I've given here only applies to totally formalized, computerized mathematical theories. But Gödel doesn't think it applies to human mathematicians.

Gregory Chaitin:

But to be able to say what you can prove and show that there are limitations, you have to give a very precise definition of the methods you're allowing in the proofs. And once you do that, you're in trouble, because if it is n bits of methods that you're allowing for mathematical proofs, then elegant programs that are more than n bits long, you're not going to be able to prove that they're elegant. Does that sound more understandable?

Robert J. Marks:

Yeah. No, so it's a proof by contradiction. You assume that the elegant program detector was algorithmic, and then you showed that there was a contradiction in your assumption.

Gregory Chaitin:

Well, it's-

Robert J. Marks:

Therefore, it can't exist.

Gregory Chaitin:

Right. Nobody laughed, so I guess I fumbled the ball.

Robert J. Marks:

Oh. Nobody laughed. Okay. What was the joke? I missed it?

Gregory Chaitin:

Well, it's the contradiction.

Robert J. Marks:

Oh, okay. The contradiction. Okay. Maybe it was because of my familiarity with the proof.

Gregory Chaitin:

I explained it badly.

Robert J. Marks:

No, no. I don't think so at all. I think that in my class when I explain proof by contradiction, let me tell you my favorite example. It's the proof that all positive integers are interesting.

Gregory Chaitin:

That's related to all of this stuff that-

Robert J. Marks:

It is. It is. And so you've heard about this?

Gregory Chaitin:

Yeah.

Robert J. Marks:

So the idea is that if they are interesting, you assume the opposite. If they're not interesting, that there is a smallest, non-interesting number, but hey, that's interesting.

Gregory Chaitin:

Absolutely.

Robert J. Marks:

That's the proof by contradiction.

Gregory Chaitin:

That’s sort of similar, because interesting number, if you want to define it carefully, mathematically, is one for which there’s a program to calculate it that’s smaller than it is, if you want to have a concrete notion of interesting. So an uninteresting number would be one whose numerical value is irreducible. That’s probably better than what I was saying. Yeah, you’re right. That’s a good way to explain this.

Gregory Chaitin:

And one good definition of interesting is an interesting number is one that stands out, because there is a more concise definition of it or, more precisely, a program that is substantially smaller than its numerical value that calculates it… that’s some way it stands out from the run-of-the-mill numbers. And the run-of-the-mill numbers are ones whose numerical value is an incompressible or irreducible string of bits. So you can sort of go step by step from that paradox about the smallest uninteresting number, which is, ipso facto, interesting, to a proof of an incompleteness result very similar to mine, merely by considering provably interesting numbers.

Robert J. Marks:

Okay. Very interesting. Okay. New topic. I wanted to talk to you about... We're talking just in general about the unknowable. Roger Penrose, he recently won a Nobel Prize for his work with Stephen Hawking and black hole theory, wrote a book called *The Emperor's New Mind*, and he had a had a follow-up, which I think is *The Shadows of the Mind* or something like that. But in the book, he says that "creativity is non-computable." He uses your work, along with Turing's work and Gödel’s work, to make an argument that creativity is non-computable and therefore is something which, if we're to pursue it by artificial intelligence, something that will never be done. Now he refers to quantum as possibly the only non-algorithmic thing which occurs in nature. So he says all of this stuff must be due to, he posits of quantum tubules in our brain and quantum collapse or something like that. And I'm wondering if you have any thoughts on whether creativity is computable or not computable or do we know yet?

Gregory Chaitin:

I do have thoughts on this, and one or two remarks. One remark is that I did meet Penrose at Cambridge at a meeting. We were both speakers, and I went to him and I said, "Is the reason you think that a machine can't equal human intelligence, because you believe that we have a divine spark and computers don't have a divine spark?" And he answered a trifle annoyed, I think, "Not at all."

Robert J. Marks:

Yes.

Gregory Chaitin:

So-

Robert J. Marks:

Well, he did revert to a materialistic solution, which is quantum-mechanical, so yes.

Gregory Chaitin:

I'd been worried a lot about creativity lately when I started working on biological creativity and I connected it with mathematical creativity. There is this paradox just like the first uninteresting number is, ipso facto, interesting, there's a problem with creativity with having an algorithm for creativity, a computer program for creativity. The problem is that if you know how to do something, ipso facto, it's not creative anymore.

Robert J. Marks:

Yes. That's the problem with identifying creativity. One can have a creative spark, and you explained it to somebody, they sit there and rub their chin and said, "Well, that's obvious."

Gregory Chaitin:

Well, creativity is what we don't know how to do. And so, it looks like it's a hard thing to program, because if we try to program creativity, well, that just becomes something mechanical and the frontier between what's creative and what isn't just moves a little forward. It's a problem. I think that my attempts of a little toy model of biological evolution - this is a controversial point - is a first step in the direction of a mathematical theory of creativity. I believe that Gödel's incompleteness theorem and Turing's work on the uncomputability of the halting problem are baby steps - well, they're big, big baby steps - in the direction of a theory of creativity. That's normally not how you view them, but I feel they feed into my little attempts at looking at biological creativity with a painfully simplified toy model.

Gregory Chaitin:

So there is a paradoxical aspect to creativity. Now, you could have a mathematical theory of creativity that enables you to prove theorems about creativity, but is not implemented in software, that doesn't give you an algorithm for being creative. Because if it's an algorithm, it's not creative, right? But you might be able to prove theorems about creativity, you see? Like I can prove theorems that most numbers are random or unstructured. I can't produce individual examples that I'm certain are. So it might be that you could prove theorems about creativity, but the theory wouldn't give you a formula, a recipe, for being creative. Because once it does that, then it's not creative, you see? There's this paradox.

Robert J. Marks:

And also, those theorems that you're talking about are kind of meta. You're using creativity to write theorems about creativity.

Gregory Chaitin:

Of course. Absolutely.

Robert J. Marks:

Yes. And one of the important things is to define creativity. I know somebody that knows you, I'm not sure if you know him... Selmer Bringsjord, who said he met you, I believe, at a recent meeting. But he has something called the Lovelace Test, which is a lot better than the Turing Test. He says that a computer program will be creative if and only if that computer program does something which is outside of the intent or explanation of the programmer. And I think that that's a very... The Lovelace Test is one that I believe that hasn't been passed yet. So it's going to be interesting to see if in the future of AI, that we do have anything creative.

Gregory Chaitin:

Well, I think I met this gentleman in Thessaloniki.

Robert J. Marks:

Yes. I believe that was where he mentioned that he met you. Yes.

Gregory Chaitin:

Well, Turing says he will believe that a computer is intelligent if the computer will punish him for saying that the computers aren't intelligent.

Robert J. Marks:

Okay. Now see, I think that's funny. Okay. What did he mean by punish you? Come up and whack you across the head?

Gregory Chaitin:

Probably.

Robert J. Marks:

What'd he mean by that?

Gregory Chaitin:

I guess Turing is referring to a notion of truth based on political considerations. People will say something's true if society will fire you from your job if you disagree.

Robert J. Marks:

The other thing you mentioned in your book is Tononi's Phi function, model of consciousness, which I must admit I don't totally understand. But in-

Gregory Chaitin:

Me neither.

Robert J. Marks:

Don't you? Okay. Well, that's good.

Gregory Chaitin:

It's complicated. It's complicated.

Robert J. Marks:

It is complicated. And in Christof Koch's, one of his presentations about the theory, he got the people in Silicon Valley mad, because this was a report that I got from somebody that attended Koch's lecture, that they were mad that this was possibly non-computable, at least with the resources that we have now or the immediate future. So I don't know. It seems to me that there's... Who was it? It was Stephen Hawking that says, "Nothing is ever proven in physics. You just accumulate evidence." And so, I think evidence is accumulating in so far as the non-computability of some human attributes. At least that's my personal take.

Gregory Chaitin:

I had something I wanted to say.

Robert J. Marks:

Okay. About Tononi?

Gregory Chaitin:

Tononi’s Phi has a complicated definition and you would need to do immense amount of computing. You have to look at all possible partitions of a physical system and calculate certain mutual informations… It’s a horrendous exponential growth computation. I know Tononi’s Phi is fashionable now, but I actually prefer the original approach in Chalmers’ 1996 book, *The Conscious Mind*. He has an idea similar to Leibniz’s monads of panpsychism.

Robert J. Marks:

Oh, yes.

Gregory Chaitin:

Everything has some degree of consciousness. It may be greater. It may be smaller. The maximum monad corresponds to God, whose consciousness is the largest possible conscious of everything, right? A rock doesn't have that much consciousness. But he believes a physical system has *n* bits of consciousness if it has *n* bits of memory and processes these *n* bits. So that would mean that an on-off light switch would have one bit of consciousness, and a human being would have a lot of bits of consciousness. It's hard to have a cutoff. For example, if humans are conscious and the people who love dogs are certain that dogs are conscious, so is there a sudden place where consciousness blanks off as you go to more primitive life-forms... bacteria, viruses, light switches? That looks a little implausible from a philosophical point of view. It seems more like it'll just be gradually less and less consciousness, right? And you can go in the other direction. You can have a corporation. Does that have consciousness? Does the internet have consciousness? Does the whole universe have consciousness, which presumably would be God?

Gregory Chaitin:

My latest and hopefully not last paper is on consciousness, and it's just being published in a book in honor of one of my distinguished colleagues here in Brazil, but it's on my website. It's "Consciousness and Information, Classical Quantum or Algorithmic?" Because Chalmers didn't know – maybe at the time, there weren't – at least three definitions of information. There's Shannon information, entropy. There's quantum information theory, which maybe didn't exist in 1996 or was very incipient and now it's a very big, fashionable topic. And there's algorithmic information. And I'm sort of looking at Chalmers, is it 25 years later? So it's a philosophical essay.

Robert J. Marks:

It's interesting. I've always looked at panpsychism as kind of a weird philosophy. I'm wondering if there's any way that it can be tested. I doubt it. It's going to be interesting to see if it can. But the posit is that consciousness is part of the universe just like mass and energy and all of the other stuff.

Gregory Chaitin:

Well, they're idealistic philosophies, which say that the universe is spirit really and matter is an illusion. So that's sort of related to an idea that I've been backing, which is the universe is made from information, that that's the basic ontological basis. The normal view if you dabble in metaphysics is the universe is made from mathematics. That's a Pythagorean idea, that all is number and God is a mathematician. But I prefer to say that all is algorithm, God is a programmer.

Gregory Chaitin:

There's a book by a theologian, an Italian theologian, a priest, on this subject called *Bit Bang. La nascita della filosofia digitale*. It's a wonderful book. Unfortunately, it's only in Italian. So saying that the universe is built out of information is like saying that the universe is really built out of spirit or the universe is in the mind of God, it's not a material object. And the new version that physicists love is to say the universe is built out of quantum information. They want to try to get everything out of quantum information, including spacetime due to entanglement between qubits, for example. That's a fashionable topic. But in a way, where we're looking at an old idea, which is that the universe is in the mind of God or the universe is spirit and matter is secondary. It's idealism as opposed to materialism, which is why this theologian was interested and put together a wonderful book surveying all of this work. But unfortunately, I don't think anybody's translated. Maybe Google Translate could do a great job on it.

Robert J. Marks:

Yeah. Yeah. Maybe. Yeah. I think that still has a long way to go. Well, Greg, we're running really over time, so we're going to have to end this conversation. So thank you so very much. This was just a fascinating chat.

Robert J. Marks:

We've been talking with Professor Gregory Chaitin about knowability and unknowability and what that means and the implications for us. Fascinating stuff. So until next time on Mind Matters News, be of good cheer.

Announcer:

This has been Mind Matters News with your host Robert J. Marks. Explore more at mindmatters.ai. That's mindmatters.ai. Mind Matters News is directed and edited by Austin Egbert. The opinions expressed on this program are solely those of the speakers.