The T Test

Dexter Kim

A: So what's minimal complexity anyway?

B: It's ka-naught. The complexity of the algorithm that would pass the Turing test that can code an algorithm that would pass the Turing test.

A: ... I'm sorry, could you repeat that again?

C: (sigh) I'll explain. You'll probably know about the Turing test.

A: Of course.

C: Now, there are many ways to circumvent what Turing test wants to evaluate. You really cannot think that chat-bots are genuinely intelligent, right? That is where minimal complexity kicks in. It requires the algorithm to pass the Turing test AND write an algorithm that will pass the Turing test.

A: Oh...

C: By the way, what's ka?

B: I believe it's the first letter of the Korean alphabet or something. The concept originated from a Korean guy, if I remember correctly.

A: So you mean there are ka-one, ka-two and so forth. Wow, I'd call that a bad habit from Cantor.

Georg Cantor is a German mathematician that used aleph, the first letter of the Hebrew alphabet, to denote cardinality of infinite sets.

B: Well, it does resemble the aleph numbers of Cantor.

A: Anyway, how are the numbers arranged?

B: ka-naught equals minimal complexity. ka-one equals the complexity of an algorithm that passes the Turing test and writes an algorithm of complexity ka-naught. ka-two, ka-one. ka-three, ka-two. You get the idea.

A: Well, sort of. And you're asking me for the upper limit of minimal complexity?

B: Yep. Any idea?

A: I do, but I don't see any point in doing that. We haven't a single algorithm that passed the test, you know.

C: Well, let's be optimistic on that.

A: This means even God hasn't reached ka-naught! What's the point?

B: The point is that we'll be in trouble if we do not publish anything.

A moment of silence has passed.

A: Too bad I'll have to agree on that.
C: So, what's your idea?
A: Since you said 'let's be optimistic', let's just suppose that such an algorithm will be built in around a thousand years.
B: I think you're underestimating the power of humanity, but go on anyway.
A: Now, humans have been around this planet for like hundred thousand years, so adding another thousand won't change anything. Let's take that as the running time of the algorithm.
C: Seems legit, but what's the point?
A: We can use this number to calculate the upper bound on number of computations.

A pause requested more explanation.

A: There is a limit called Bremermann's limit. It's a physical limit on computation speed, albeit theoretical. It's around ten to the power of fifty bits per second.
C: That's huge.
A: Yeah. Anyway, applying this to the Earth, we have about ten to the power of seventy-five bits per second.
B: Wait, didn't you say fifty?
A: Oh, it's fifty bps per kilogram.
B: So it depends on the mass of the computer?
A: Yep. Anyway, a hundred thousand years of running time means any algorithm that was worked out on Earth cannot be more complex than ten to the power of eighty-eight. This would be the most conservative estimate of the upper bound on minimal complexity. If we can build one, that is.
C: So what we have calculated is in fact the maximal complexity.
A: Well, sort of.
B: I've got a question. Does the limit apply to all computers or computers that are quantum?
A: It should apply to all computers, I think. Ask the guy with a physics degree.
C: You don't normally teach that in physics courses. Anyway, the estimate is too crude. I'm not sure whether if it would qualify as a paper at all.
A: Yeah, I do agree that it rather suits a fiction. We could cut down the numbers, but I think it would be hard to get past sixty.

B: Wait, when did eighty-eight become a sixty?

A: Take the mass as one human brain weight, running time a lifespan. That gives a magnitude around ten, and fifty plus ten gives sixty.

C: It's still enormous. So that's the estimate you can give us?

A: For now. Maybe I'll be able to cut down the number -- around forty-two, for example -- but I'll have to think about it.

B: Alright. Let's call it a day then.

A: Let's call it a day.

C: Okay. Call it a day.

The first voice has logged out.

B: So, how do you think about the algorithm?

C: I'm quite sure anyone would agree that the algorithm passed the Turing test. It must've been a hard work to make it speak ungrammatical sentences. Many people tend to forget that they speak ungrammatically.

B: Thanks. Your help helped me a lot.

C: I don't remember doing anything, you know.

B: Well, it was you who told me about synapses, wasn't it? Especially about inhibitory synapses. I have revised my algorithm based on your ideas.

C: So, how did you modify it?

B: I've got rid of Markov chains and implemented Feynman summation. Stability issues caused me some trouble though.

C: Ho, so you complexified probabilities? I never heard of anything like that!

B: Well, I adopted negative probabilities, but not complex. I don't think you can have anything like that at all.

C: Well it's true that Feynman's original integral dealt with amplitudes, not probabilities. Anyway, did my memory help?

B: N...o, not really. I had to pick out the units that did not contradict the ones I already implemented. Only a tenth of them were used.

C: Thinking about it, it's quite interesting that you've developed your own memory isn't it? I never thought memories would accumulate itself on its own. I'm rather surprised that your personality differs from mine a lot.

B: You don't need to keep reminding me that you coded me. It sometimes pisses me off, you know.
C: Oh, I didn't mean to offend you. I'll apologize if it sounded rude.
B: Nevermind. Anyway, do you think this will count as an actual calculation of ka-naught?
C: Yeap. Don't you agree?
B: Well, I have something in my mind that I cannot describe well, but I think the test is a bit too crude.
C: Will you explain?

The second voice took some time to get the ideas organized.

B: The Turing test is a test devised to test intelligibility, right?
C: Well, obviously.
B: The underlying assumption of the Turing test is that conversation is a genuinely intelligible activity.
C: You mean intelligent. Go on.
B: Ugh. Intelligent. Anyway, the ability to converse might not be the sign of intelligence. The real sign of intelligence might be the imagining of converse.
C: Imagining?
B: Well, I'd say simulating a conversation. I mean creating an imaginary conversation that has a logically meaningful structure. A chat-bot that's simply designed to choose an appropriate response cannot do that.
C: It's a good idea that we can add at the end of our paper, but we don't have enough time. The deadline is only five minutes away.
B: Oh well then. Let's just submit the draft. You've checked the numbers correctly, right?
C: I have. By the way, congratulations for becoming the first algorithm to author an academic paper.
B: Thanks. See you at the conference then.

I stopped the simulation and turned around.

"I think this is a constructional proof that ka-one equals ka-naught. What do you think?"
"Well, I'm not so sure about the minimality of the algorithm. Does this algorithm satisfy the lower bound?"
"You mean the lower bound of ka-naught? I'll have to think about that."

Oh well. It seems my graduation is still far from reach.

"Anyway, I've changed my mind after looking at your simulations. It seems likely that universality hypothesis is true."
"I would have agreed if I hadn't troubled myself of rewriting the algorithm from scratch."
"But the length of algorithm didn't change much, did it? Do you think you can shorten it further?"
"It's impossible. I never thought revising a ka-naught into a ka-one would be this challenging."
"But that's what you need to do to graduate. Oh, I'm a bit late, so excuse myself."
"I'll see you tomorrow then"

My advisor left the office. Since I'm all worn out after working around the clock for days, I'd better go find myself a place to take a nap. I never would have overworked if I knew my advisor would be this unimpressed.

[End of simulation. To start a new simulation, enter a new keyword. (Converse generator ver. 0.577)]