

Re: Turing machines, quantum computers, and alephs

Source: <http://sci.tech-archive.net/Archive/sci.math/2005-03/7643.html>

From: Gerry Quinn (*gerryq_at_DELETETHISindigo.ie*)

Date: 03/22/05

Date: Tue, 22 Mar 2005 10:16:21 -0000

In article <1111453497.964842.107290@o13g2000cwo.googlegroups.com>, hack@watson.ibm.com says...

> *Gerry Quinn wrote:*

[I replaced this paragraph because it is relevant to what you say below]

> *Surely that depends instead on whether the initial settings of the qubits are finitely describable? The Turing machine has an infinite number of tape segments, but if it starts with a blank tape, or a tape containing a certain pattern a million bits long, it is finitely describable even though the tape has infinitely many zeros. Likewise, the abstract quantum computer with infinitely many qubits might start off with all but a million in a standard initial state – in which case it seems no less finite in your sense than the abstract Turing machine with its infinite but mostly zeroed tape.*

>> *ATM (abstract Turing machine with an infinite tape – what is meant by the generic term 'Turing Machine' – an abstract concept that could not be built in the real world.)*

> *Common confusion between infinite and unbounded, I'm afraid. The critical information that is missing in the summary description above is that, if one uses the "infinite tape" metaphor (as opposed to the usual "unbounded tape" one), one must also state that the initial state of the "infinite" tape has only finitely many marked cells. (On a two-symbol tape, the initial state has a finite number of 1-bits, for example.)*

Observe the top paragraph, which was part of the post you replied to, and which I put back in. I don't disagree with this point. The abstract quantum computer, that is to a physical quantum computer as a Turing machine is to a physical computer of the same kind, must have only finitely many qubits starting in a non-default state. And they must be a finite 'distance' from a given starting qubit – 'finitely describable' isn't quite right because it doesn't include the last part.

[Maybe there is a problem with the abstract quantum computer here. Do we have to set up entangled qubits in advance, or is it no different from

segments of a tape, which of course are physically entangled too? But if so, surely the 'incremental' approach I describe below can solve it?]

[Incidentally, "turing machine infinite tape" has 28800 hits on Google. "turing machine unbounded tape" has 10500. And Turing said "infinite". Hits on Google don't prove much, but if you want to assert that 'unbounded' is more accurate than 'infinite' (which I don;t believe it is, really – it seems evasive more than anything), it does not help to make dubious claims that to use 'infinite' is somehow unusual.]

A Turing machine cannot be built in the real world. Do you disagree with this? If not, how can you build one that will not run out of tape given some starting setups?

If we are to allow incremental approaches to the problem, why don't we compare incremental series of mini-Turing machines with finite tapes, versus incremental series of quantum computers with finite qubits? If either balks at completing a given calculation, we can try again with n extra tape segments, or n extra qubits, until and if a given problem has a solution. Maybe that makes a bit more work for the TM, which we can imagine could be extended using a tape splicer, leaving current calculations intact. But that seems like a technicality, given that it gets there anyway, and we only multiply the work by an finite (and non-exponential) factor, assuming it halts.

> *If an infinite initial marking is allowed, the machine becomes strictly more powerful (rises in the Turing-degree hierarchy). For example, one could solve the Halting Problem for all ordinary Turing machines by table lookup!*

Interesting perspective, but not what we're talking about here. And I don't believe it is correct, anyway, because of EXACTLY the same error that I believe I have uncovered, and which provoked this thread.

If you have marked up the solution to the halting problem for all Turing machines up to a certain finite descriptor size, your tape is finite, and its length is an exponential function of the descriptor size. But if you let the descriptor size rize to infinity (Aleph-0), as you must if you are to cater for "all ordinary Turing machines", then the length needed for the marked tape must rise to the order of $2^{(\text{Aleph}-0)}$ or Aleph-1. And Aleph-1 marks won't fit on a tape, even an infinite one, because it only has room for Aleph-0 symbols.

> *Since any accepting computation is finite by definition, if the initial tape has only finitely many marked cells, so do all other states encountered, and hence the same computation could have been carried out with a finite tape. So the two tape metaphors ("infinite" and "unbounded") are equivalent in this case.*

But that applies only to programs that halt. And if all programs halted, we wouldn't have anything interesting to discuss here. If you

define a computation as the result of a program that halts, you can't tell which programs will yield computations. Not with a finite tape. You can, of course, make a list of halting programs that require such and such a length of tape, then a supplementary list of those that halt if given that length plus one, and so on.

Surely if you want to go this route ('finitising' the abstract Turing machine), you can do it with the abstract quantum computer too, by adding one qubit at a time? The Turing machine can't escape the exponential difference between it and the qubit computer – not if it stays finite, and not if it runs to infinity. It seems to me that the claim that they are equivalent is dependent on arguments that slide invalidly between one and the other.

– Gerry Quinn