

Re: Arthur O'Dwyer on the feasibility of simulating a Turing Machine

Source: <http://coding.derkeiler.com/Archive/General/comp.programming/2004-02/2605.html>

From: Edward G. Nilges (*spinoza1111_at_yahoo.com*)

Date: 02/29/04

Date: 28 Feb 2004 17:29:23 -0800

Calum <calum.bulk@ntlworld.com> wrote in message news:<c1qimg\$eed\$1@news5.svr.pol.co.uk>...

> *Edward G. Nilges wrote:*

> > *Willem <willem@stack.nl> wrote in message news:<slrnc3v69e.9cf.willem@toad.stack.nl>...*

> >

> > > *This has already shown to be false; there's only a finite amount of matter*

> > > *in the universe.*

> > >

> > > *Edward wrote:*

> > >

> > > *However, do we know if there is a finite amount of time? For by simply*

> > > *having the simulator act as a sort of virus we can exchange raum for*

> > > *zeit, space for time.*

> > >

> > > *Each operation of any TM-like machine increases the entropy of the universe*

> > > *by some amount (with a fixed nonzero minimum), therefore an UTM needs to be*

> > > *able to increase the entropy of the universe by an unlimited amount.*

> > > *But a finite universe has a finite maximum entropy.*

> >

> >

> > *In fact, this is false. For any calculation that halts, the universal*

> > *turing machine will execute necessarily a finite number of operations.*

>

> *But, you can always devise a computation that requires more*

> *time/energy/space/memory/tape than the universe is capable of. Even one*

> *that halts.*

If the universe goes on indefinitely this is wrong.

>

> *E.g.*

>

> *bool array[NUM_PARTICLES_IN_UNIVERSE+1];*

> *for(int i=0; i<NUM_PARTICLES_IN_UNIVERSE+1; ++i)*

> *array[i] = is_prime(i);*

>

> *Your assertion that any desktop PC can compute anything is of course*

> *ridiculous.*

comp.programming: Re: Arthur O'Dwyer on the feasibility of simulating a Turing Machine

The assertion is that given enough time and memory space, a desktop PC can compute anything. This assertion is true and equivalent to Church's thesis re the Turing machine.

>

> > *All you're affirming is that BOTH the simulator and the "ideal" will consume the universe if they loop. The UTM and the simulator are in fact exactly equivalent in this regard.*

> >

> > *The UTM, *qua* UTM, has no special permission to continue looping in the formalism merely because it is either a Turing Machine or a Universal TM: to think so is to confuse the designation TM or UTM with some sort of honorific which (again) somehow raises Turing's mere contrivance ABOVE writing and into speech, where the UTM is allowed to loop.*

> >

> > *In fact, the formalism is silent on a number of issues including how the tape is expanded, how long the computation will take as a cardinal number and not a function of its input, and here, whether the machine in a nonhalting state is permitted to loop.*

>

> *Of course, it is important to realise that a UTM is a thought experiment, a mathematical entity. Its purpose is primarily to discuss the mathematical nature of computation. The point is it DOES NOT MATTER how the tape is extended, or whether it is performed by a computer or by pencil and paper. That's what's nice about mathematics, you always get the same result.*

>

This is true so far as it goes. But the description of the TM needs to be rephrased depending on your philosophy of mathematics.

The TM is an "entity" only to the Platonist who ground the truth of mathematics in the existence of Forms. To an Intuitionist or Marxist the TM is a text and a set of construction instructions.

> *And as for "how long the computation will take", does the phrase "Halting Problem" have any meaning to you?*

Yes, and it doesn't apply, since this NG is collectively dazzled by HALTING Turing Machines that take up all the matter in the universe for the deconstructive reasons I've explained in the following paper:

"...writing is not only an auxiliary means in the service of science—and possibly its object—but as Husserl in particular pointed out in *The Origin of Geometry*, the condition of the possibility of ideal objects and therefore of scientific objectivity."

I do find this discussion of "the feasibility of simulating a Turing Machine with actual software" interesting because it shows how Jacques Derrida's so-called deconstruction applies to a critical theory of computer science.

comp.programming: Re: Arthur O'Dwyer on the feasibility of simulating a Turing Machine

For as Ian Smith claims, a consensus seems to be that no "Turing Machine simulator exists that is a full simulation of a Turing Machine".

This seems wrong, or wrongly phrased, for there is no one TM. For starters, there exist in the theory multiple machines, categorized into two sets, one large and one "small". The first is the set of all special purpose TMs, armed with a tape that doesn't perform the calculations of a Universal Turing Machine, which is able to simulate any other TM including UTMs when its tape contains a suitably encoded list of the quintuples of the simulated machine.

We can see pace Ian that most special purpose Turing Machines can be simulated as long as their computation and tape is bounded, and known to be so. Furthermore, any UTM that contains instructions to halt when its tape length or computation time approaches a limit can be simulated.

Now, I concede that a computer simulation of a Turing Machine may fail on storage limitations for arbitrary TM. But this (empirical) fact fails to prove that "it is not possible to simulate a TM with software".

I've already shown, as against the rather simple-minded objection that "the simulator will run out of memory at some point for some computation", that to address this problem, all you have to do is (1) assume that the universe and the Internet are from this day immortal and (2) implement the simulator not as a program running on a single machine but as a computer virus.

But there are two deeper issues here than such contrivances...such conveniences.

The first is that one, and only one, completely unexamined philosophy of mathematics dominates the discourse cascade, and, as far as a philosophy can be discredited, this philosophy is discredited.

This is logicism, or Platonism, which baldly assumes that mathematical objects have reality independent of the mind, exist in a World of Forms and furthermore that real infinities exist. It was refuted by the paradoxes, Godel's result and in general Bertrand Russell's failure to derive mathematics from logic.

Completely unexamined is the intuitionistic alternative which based on Kant posits mathematical reality as "real" but so embedded in experience as not to be meaningful once abstracted from experience.

The Pop, ignorant reading of Kant is that in so many tedious words, Kant showed that ultimate reality is unknowable and, of course, Kant did nothing of the sort. Instead he actually takes the patient reader of the *Kritik der Reinen Vernunft* of the only ultimate reality

accessible to experience, and this is the way experience necessarily takes place in space and time.

In this tour, entities outside of experience seem to emerge like Hamlet's father from the general murk of Kant's style, but like ghosts at cock-crow they are dispelled by Kant himself who shows that language adequate to ordinary experience is inadequate to describing the abstraction itself.

In mathematics, a succession of triangular shapes suggested to Euclid a common three-sided figure which when sufficiently abstracted by the necessary operation of memory and anticipation was seen to be at once "perfect", and unattainable, even on a computer screen: and one reason for the vicious return of a barbaric Platonism (which doesn't deign to even consider alternatives) is that software presents what seems to be (but is not) the Form.

The Platonist hypostatizes the Turing Machine as existing, in a fashion which "can't be simulated by a mere program", in a World of Forms, and one sense in the consensus here that the very idea of "simulating" a Turing machine, even for such a humble task as teaching, is a form of farting in the Church of Reason.

The Intuitionist regards the TM instead as a set of instructions for building something (zuivere beelding) which unlike a house is without apriori or aposteriori bounds. Thus the "scope" is the same as experience itself. But he doesn't imagine for a minute that TMs already exist in God's computer room and for this reason the intuitionist would probably be friendlier towards the idea of simulating a Turing Machine.

He would demand, however, that the Turing Machine not preallocate its tape as TM_TAPE tape[1024], or any fixed number but would demand instead that the tape be expandable UNTIL malloc failure (to speak in the barbarisms of the C language).

The Intuitionist would not I think infer from the TM's mortality any failure to simulate a "real" TM.

For note that the "real" TM doesn't have an unbounded tape. It merely has the ability, deliberately unspecified, to demand more tape. A "real" program simulates this ability when it aborts on not enough storage and thus demands that its programmer expands its arrays and/or installs more memory.

More strongly a Marxist philosopher of mathematics would say that what Turing actually did was to abstract from the praxis of actual working people a model of what it means to mechanically compute, whether by watching clerks at work or observing simple office machines of the 1930s in action.

But be that as it may, Ian reaches an absurd conclusion, that computers cannot simulate Turing machines. The conclusion is absurd because it denies Church's Thesis, that Turing Machines express the limits of what is computable.

But we must explore more deeply why this absurd conclusion becomes the ng consensus and the surprising source of the glimmer of understanding is Jacques Derrida's *Of Grammatology*.

This extraordinarily difficult book is a meditation on an unnoticed theme in Western philosophy, expressed in a simple and simple minded fashion as the privileging of speech over writing. But this precis needs to be expanded.

It seems that as a member of the generation of the 1960s Derrida was troubled by the way in which so many learned disciplines, including linguistics and philosophy, tend to cut away the ground on which they stand at a key point, and this gesture becomes the central mystery of the discipline...one which is maintained in a terroristic fashion because it can always be used to ensure the subordination of truth to power in the West.

Thus in both theoretical linguistics and the practical teaching of language the need to address the spoken word is so foregrounded that it is claimed that written symbols actually represent speech, a claim which isn't argued for as much as it is axiomatic. Thus in the practical teaching of language the students are rather overwhelmed with written texts which use grammatical tables and illustrations of the spoken word only to be told both at the start of the class and at its end that of course, the living, vibrant, spoken language is never mastered by such contrivances but by dwelling among the peasants themselves.

At one and the same time, an offer is made and rescinded in a gesture which Derrida was the first to notice.

This gesture doesn't seem to be repeated by other cultures of writing. It appears that at least traditionally in China, no idea existed that speech was logically prior to writing and as in Moslem cultures, the traditional high regard for literacy, coupled with its absence, means that "more writing" was naturally prized, without second thoughts. In fact, the more recent emphasis on China on language change and reform, and the literal attack on cultures of the book in the Cultural Revolution may be due to western influence, and part of the tension between the West and Islam may be due to the way in which incompetent Western translators and journalists translate written metaphors as vulgar spoken attacks.

Derrida speaks in terms of "oppositions" which structure Western thought such as "the Letter killeth but the Spirit giveth life", which are undercut because the very text is transmitted not through the

Spirit but through the Letter.

I think in this discussion an erroneous result is arrived at (a Turing Machine can't be simulated) because of the completely unconscious operation of the Western binary opposition between speech and writing.

A mere intuitionistic construction rule is merely a piece of writing, a set of instructions and a story which the reader can then use, in an unbounded fashion, to create new Turing Machines: just as the reader of Stephen Wolfram's *A New Kind of Science* can use his related formalism in an unbounded fashion. Wolfram while claiming that his automata may be useful in physics certainly doesn't maintain that you cannot simulate his automata on a computer for the very good reason that he would not have been able to write *A New Kind of Science* without writing simulators.

But note that in this way the binary opposition so operates that the abstract Turing Machine becomes "unwritable" in the sense of not being simulated and the (literally false) claim is a command masquerading as a truth value. Just as Aristotle had no good reason to say that written symbols "symbolize" spoken sounds but said it anyway because it simplified his task and policed the boundaries of his speech, a division is maintained between theory and practice in mere computer science between mathematical abstractions, which are immortal, it seems, and are best spoken of, best whispered about, best muttered about as Eleusinian mystery by adepts and not written about...and not "simulated".

Derrida remarks that this gesture, which constructs the sociological foundations of many disciplines, entails that the scientist, the linguist in the case of de Saussure, drop the mask of scientific courtesy and objectivity because he is, without being able to acknowledge it, policing the frontiers of the grounds of the Palace of Reason. Thus violations, including written programs which "simulate a Turing machine" become not only incorrect but wrong, pernicious and evil, monstrum horribilum which like Carthage delenda must have to be.

Derrida has long interrogated the prescientific and barbaric pose struck by he who like the Heathfield man, Rousseau or de Saussure would set himself up to patrol the boundaries of a discipline...by making sure that the very technology of the discipline is not used in full and for the purposes intended. Thus deSaussure rages in the *Course of General Linguistics* against orthography and gives secret satisfaction to boneheads in many lands who are actually proud of an inability to spell, while they condemn mispronunciation.

Derrida was struck in his exchange with John Searle (who I learned recently was a soured survivor of the wars in the 1960s at UCB) by the lack of affect and humor which must preserve boundaries even at the cost of common sense: even at the cost of a regression to barbarism (as when the Heath man tells us that programming, has nothing to do

comp.programming: Re: Arthur O'Dwyer on the feasibility of simulating a Turing Machine

with programmers, being left by security men weeping to their cars in a layoff let us say).

Indeed, it was only at the "better" universities that I perceived at all any interest in such contrivances and the demand for "practicality" at inferior universities has the effect of separating theoretical constructs from working individuals.

The Turing Machine becomes speech to software writing. On the side of the TM is hypostatized the academy, Truth, and disinterest in commercial gain. On the side of software is cast the demons of commerce and the disabling nature of writing which is assumed without argument to cripple its adepts and make them at best a convenient, but disposable, class of scribes who write down the commands of kings.

Thus the discourse cascade rages against the "simulation" of the TM in precisely the same way Plato raged against representational art and writing.

Software winds up with a bad or divided conscience in the West because at one and the same time it must maintain the superiority of self-present speech while it continually undercuts self-presence by multiplying forms of writing. For example, the completely narcissistic claim has to be made that "my protocols are 'clearer' than thine" in economic competition which when translated to the terms of the binary opposition means that "my" protocols better serve the seen need for openness and transparency.

The paradox is that in so moving towards that stated goal, the protocols and standards become ever more opaque to end users and to people considered as pure victims of operational systems.

Thus it is claimed that "my" Java Beans are better encapsulators of business rules than your .Net but in both cases one will search the actual data base in vain for the rules which are hidden from the customer and supplier and "protected" against change or even understanding by the CEO...who has now in America to nonetheless sign a statement, risking prison, if his data processing people have misused the hidden business rules (cf Sarbanes-Oxley).

Derrida has no "cure" for the ailment he describes for he so describes the ailment as (like a bug in a compiler on which the users now rely) so constitutive of Western thought that to cure the ailment would kill the patient. And in the case of software, the writing has fulfilled Plato's worst fears, for his worst fear was that the writing would take on a bad life of its own and destroy its host.

This seems to be happening because precisely as the West globalizes Western ways using the Internet, the consumers regurgitate the content in their own way: Chinese shall be the dominant language on the Internet by 2008.

Re: Arthur O'Dwyer on the feasibility of simulating a Turing Machine