# NON-COMPUTABILITY OF HUMAN INTELLIGENCE

YASHA SAVELYEV

Abstract. We revisit the question (most famously) initiated by Turing: Can human intelligence be completely modelled by a Turing machine? To give away the ending we show here that the answer is *no*. More specifically we show that at least some meaningful thought processes of the brain cannot be Turing computable. In particular some physical processes are not Turing computable, which is not entirely expected. The main difference of our argument with the well known Lucas-Penrose argument is that we do not use Gödel's incompleteness theorem, (although our argument seems related to Gödel's) and we do not need to assume fundamental consistency of human reasoning powers, (which is controversial) we also side-step some meta-logical issues with their argument. The argument is via a thought experiment and is at least partly physical, but no serious physical assumptions are made. The only "interesting" assumption (Hypothesis 2 in what follows) is very difficult to deny. We may moreover reinterpret the thought experiment as an actual experiment that may in principle be carried out in the future, this is also described.

We study the following question:

*Question* 1. Can human intelligence be completely modelled by a Turing machine?

We will give a complete definition of a Turing machine after the introduction. An informal definition of a Turing machine [1] is as follows: it is a machine which accepts certain inputs, and produces outputs. The outputs are determined from the inputs by a fixed finite algorithm (in a specific sense).

In particular anything that can be computed by computers as we know them can be computed by a Turing machine. For the purpose of this article the reader may simply understand a Turing machine as a digital computer with unbounded memory running a certain program. Unbounded memory is just mathematical convenience, it can in specific arguments (also of the kind we make) be replaced by non-explicitly bounded memory.

Turing himself has started on a form of Question 1 in his Computing machines and Intelligence, [2], where he also outlined an obstruction to a yes answer coming from Gödel's incompleteness theorem. He pointed out that one way to avoid this obstruction is to reject the assumption that humans are fundamentally consistent. (What the latter actually means in practice is subject to a lengthy discussion, we need some qualifier like "fundamental" as even mathematicians do not assert consistent statements at all times.)

There are a number of ways of interpreting Question 1. Turing himself was mainly interested in whether a Turing machine can fool an experimenter into believing that it is a human subject, in various Imitation Games, see [2] for examples. As this author understands, Turing believed roughly the following, (here for exposition's sake we take the liberty of compressing, perhaps not completely accurately, Turing's ideas into a neat hypothesis):

**Hypothesis 1** (Turing's hypothesis)**.** *For every given human experimenter, an imitation game, and given some bounded amount of time, one can construct a Turing machine that will fool him in this imitation game, for that amount of time.*

This particular hypothesis may well be true, (nothing in the present note contradicts it). However our interpretation of the question is a bit different. We are asking is whether there is a meaningful mathematical difference in operation of a human mind and a Turing machine. While the significance of this version of the question for computer science is perhaps arguable, for physics and biology it is profoundly important.

It should be pointed out that a common misconception is that there can be no such mathematical difference if one believes the universe to be governed by deterministic laws. We would rather not get into detailed discussion what deterministic means here exactly, since from some point of view quantum mechanics is not deterministic. Suffice to say that for our purpose here, quantum mechanics is deterministic, as the results (measurements) are at worst given by a probability distribution that is determined. But deterministic does *not* imply computable in the mathematical sense, e.g. Turing computable, (this may refer also to the above mentioned probability distribution). The latter would be computably deterministic, and this is the source of the misconception. [1]

Gödel himself first argued that such a mathematical difference exists, [7, 310]. Later Lucas [6] and Penrose [8] strongly argued that a meaningful difference exists and for a no answer to question 1. They further formalized and elaborated the obstruction coming Gödel's incompleteness theorem. And they reject the possibility that humans could be inconsistent on a fundamental level. The main common objections to their argument revolve around the assumption of the fundamental consistency of human reasoning, as well as some objections to the meta-logic of the argument, see [3] for instance. It should also be noted that for Penrose in particular non-computability of intelligence is evidence for new physics, and he has specific and very intriguing proposals with Hameroff [5], on how this can take place in the human brain. Another physical argument for non-computability is presented in Song [4].

We likewise argue here for such a mathematical difference, but we do not need any consistency assumption. We do need a certain additional hypothesis for the following but let us forget it for time being.

**Theorem 1.** *The answer to Question 1 is no. More specifically, there are human subjects that when given a very restricted role of determining in a specific sense (to be described) partially defined functions* $S : Strings \to Strings$, *for Strings set of strings in a fixed finite alphabet (in practice accepted input, and output is a natural language expression) are not Turing computable.*

*Remark* 1. Strictly speaking we prove something a bit different, we allow some data to be usable by our human $S$ that is not directly part of the input string. This is purely for convenience and the same data is made available to any would be Turing machine computing $S$. So we may just as well say that this data is part of the input string, and the only reason not to do so is that it would be unwieldy.

It is interesting to point out that as stated the theorem above does not indicate any innate superiority of human beings to Turing machines, this is not an accident, we do not prove that here. This is unlike Lucas-Penrose argument which essentially proceeds by attempting to show that for any Turing machine $M$, a particular human being can know certain truths (Gödel statements) that $M$ cannot know. (Assuming $M$ is sound as a formal system.) So Lucas-Penrose argument if it can be made to work does indicate at least some innate superiority. In our argument, the human does seem to know the truth of non-halting of a certain computation, if it exists, and the truth of non-halting may play the similar role of such a Gödel statement, but in the end we show that this computation does not exist, so this "knowledge" seems insubstantial. At any rate we cannot obviously make the same conclusion as Lucas-Penrose. That said it might be possible to adapt our argument to this purpose in some way.

As we mentioned there are certain objections to the meta-logic of the Lucas-Penrose argument, going beyond the assumption of consistency. We try to avoid meta-logical issues by introducing an "experimenter" that can effectively test whether a human subject is computed by a particular Turing machine $S'$, a kind of reverse Turing test. All logical issues should in principle disappear, although perhaps we have replaced them with physical issues, but this looks exceedingly unlikely, as the physical assumptions are very mild. The only not completely vacuous one seems to be Hypothesis 2. Following the main argument, we describe how our thought experiment can be reinterpreted as an actual experiment at least in principle, that is can in principle be carried out at some point in the future.

Although we do not use the incompleteness theorem, the basic self referential "diagonalization" argument behind it seems to lurk in the background, (like Gödel's, our argument will not be directly self referential).

[1] We should mention that there is also a notion of non-deterministic Turing machines but this "non-deterministic" is not directly related to our discussion. Moreover these machines can be simulated by Turing machines of the type we consider, so are not considered here.

As we said already, no consistency assumptions are made for our humans. For disclosure this author does believe in consistency of human beings, but to me consistency always seemed to be an emergent feature of something deeper, (consciousness perhaps).

## 1. Some preliminaries

This section can be just skimmed on a first reading. Really what we are interested in is not Turing machines per se but computations that can be simulated by Turing machine computations, these can for example be computations that a mathematician performs with paper and pencil, and indeed is the original motivation for Turing's specific model. However to introduce Turing computations we need Turing machines, here is our version, which is a common variation.

**Definition 1.1.** *A* **Turing machine** $M$ *consists of:*

- *Three infinite (1-dimensional) tapes* $T_i, T_o, T_c$*, divided into discreet cells, one next to each other. Each cell containes a symbol from some finite alphabet. A special symbol* $b$ *for blank, (the only symbol which may appear infinitely many often).*
- *Three heads* $H_i, H_o, H_c$ *(pointing devices),* $H_i$ *can read each cell in* $T_i$ *to which it points,* $H_o, H_c$ *can read/write each cell in* $T_o, T_c$ *to which it points. The heads can then move left or right on the tape.*
- *A state register that stores one of finitely many internal states of* $M$*, among these is "start" state* $q_0$*, and at least one final or "finish" state* $q_f$*.*
- *Input string* $\Sigma$*, the collection of symbols on the tape* $T_i$*, so that to the left and right of* $\Sigma$ *there are only symbols* $b$*. We assume that in state* $q_0$*,* $H_i$ *points to the beginning of the input string, and that the* $T_c$*,* $T_o$ *have only* $b$ *symbols.*
- *A finite table of instructions that given the state* $q$ *the machine is in currently, and given the symbols the heads are pointing to, tells* $M$ *to do the following:*
  1. *Replace symbols with another symbol in the cells to which the heads* $H_c, H_o$ *point (or leave them).*
  2. *Move each head* $H_i, H_c, H_o$ *left, right, or leave it in place, (independently).*
  3. *Change state* $q$ *to another state or keep it.*
- *Output string* $\Sigma_{out}$*, the collection of symbols on the tape* $T_o$*, so that to the left and right of* $\Sigma$ *there are only symbols* $b$*, when the machine state is* $q_f$ *(or one of the other final states). When the internal state is one of the final states we ask that the instructions are to do nothing, so that these are frozen states.*

We also have the following minor variations on standard definitions.

**Definition 1.2.** *A* **complete configuration** *of a Turing machine* $M$ *or* **total state** *is the collection of all current symbols on the tapes, instructions, and current internal state. A* **Turing computation** *for* $M$ *is a possibly not eventually constant sequence* $\{s_i^{M,\Sigma}\}_{i=0}^{i=\infty}$ *of complete configurations of* $M$*, determined by the input* $\Sigma$ *and the table of instructions of* $M$*, with* $s_0^{M,\Sigma}$ *the complete configuration whose internal state is* $q_0$*. If the sequence is eventually constant the limiting configuration has internal state one of the final states. When the sequence is eventually constant we say that the computation* **halts***. For a given Turing computation* $*$*, we shall write*

$$* \to x,$$

*if* $*$ *halts and* $x$ *is the output string. We write* $* \to \infty$ *if it does not halt.*

We write $M(\Sigma)$ for the output string of $M$, given the input string $\Sigma$, if the associated Turing computation, denoted by $*M(\Sigma)$ - halts.

**Definition 1.3.** *Let Strings denote the set of all finite strings of symbols in some fixed finite alphabet, for example* $\{0, 1\}$*. Given a partially defined function* $f : Strings \to Strings$*, that is a function defined on some subset of Strings - we say that a Turing machine* $M$ **computes** $f$ *if* $*M(\Sigma) \to f(\Sigma)$*, whenever* $f(\Sigma)$ *is defined.*

For later, let us call a partially defined function $f : Strings \to Strings$ as above an ***operator***, and write $\mathcal{O}$ for the set of operators.

**Definition 1.4.** *We say that a pair of Turing machines $M, M'$ are* **equivalent** *if the associated operators $M', M : Strings \to Strings$ coincide.*

## 2. Proof of Theorem 1

The reader may want to have a quick look at preliminaries before reading the following to get a hold of our notation and notions. The proof will be constructed in the form of a thought experiment. In this experiment a human subject $S$ is contained in isolation in a room, under supervision of a human experimenter E. If we restrict our $S$ to interpret and reply to certain input (as will be given below), $S$ can be understood as an operator $S : Strings \to Strings$. Suppose by contradiction that every thought process of a human being can be simulated by a Turing computation, so that $S$ is computed by some Turing machine say $S'$. Note that this involves some necessary conditions. For even if our $S$ gives answers that are unambiguously interpretable as strings, to give his answer $S$ may do intermediate steps like "check the clock" (if he had a clock) then output whatever time it is. For $S'$ to compute $S$ it has to be able translate such an action into something it can simulate, for example its pseudo-algorithm could be "I check the time corresponding to the clock available to my human" (if it is able to do this meaningfully, for example if it has access to all information that the human has access to, in particular the time of the clock usable by $S$), "then I output this time as a string". Let us summarize then.

**Definition 2.1.** *Given a string $\Sigma \in Strings$, we say that $\Sigma$ is* **acceptable** *if whenever our human subject $S$, (verbally or otherwise) given $\Sigma$, replies eventually with something that is unambiguosly interpretable as a string in Strings, (in practice $S$ just replies verbally). We then have an operator $S : Strings \to Strings$ corresponding to $S$ defined on the subset of acceptable strings. We say that a Turing machine $S'$* **computes** *$S$, if given any acceptable $\Sigma$, and whenever $S'$ is given access to all and no more information that may be used by $S$ to give his answer $S(\Sigma)$, we have $S'(\Sigma) = S(\Sigma)$.*

We shall suppose in what follows that there are no obstructions as above to computing $S$.

So we proceed, our human experimenter E is in communication with $S$, she knows the operation of a Turing machine computing $S$, and she controls all information that passes to $S$. (That is usable by $S$). We also suppose for narrative purposes that S understands natural language, basic mathematics, and basic theory of computation. $S$ has in his room a general purpose (Turing) computer, with sufficiently large memory. Here sufficiently large is so that it is enough to contain all the necessary data and complete the computation $*$ described below - if it halts, if $*$ does not halt we shall obtain a contradiction before memory can run out, if it was large enough. We will say $S$'s computer in what follows.

At this moment E passes to $S$ the following input (which we understand as one string $\Sigma = \Sigma_{S'}$):

(1) Assume that I (that is E) believe that you (as an operator) are computed by the Turing machine $S'$, whose faithful simulation is programmed into your computer. You have access to this simulation $S'$ and its source code (that is the precise specification of the operation of $S'$).

(2) You cannot check the processing speed of your computer and I can adjust the processing speed without your knowledge. (This is to avoid some pathological behavior from $S$, and logical issues. We will explain this.)
(3) If you can show that I am in contradiction you will be freed. You may use your answer to 4 below to do so.
(4) Give me an integer.

Note that instruction 3 is in a sense for aesthetics, we can run the following argument without it, $S$ will not have any motivation to proceed as follows, but the only essential point is that he *can in principle* proceed as follows, and that generates a contradiction. Moreover any ambiguity of these instructions is irrelevant, it is just a string $\Sigma$, it matters not how $S'$ and $S$ interpret it, only that a certain contradiction is deduced unambiguosly by $E$, if $S$ chooses to interpret $\Sigma$ a certain way, and that as we shall argue eventually $S$ will be found that does interpret $\Sigma$ in this way.

Now $S$ knows that the answer that is expected by E is given by the Turing computation that we shall call *computation*

$$* = *_{S'} = *S'(\Sigma).$$

As mentioned above we assume that all necessary conditions for $S'$ to compute $S$ on this string are satisfied. In practice all this means is that $S'$ can access the information on $S$'s computer that is available to $S$, and no more information than that. Since by $S'$ we really mean the simulation of $S'$ that is already on the computer of $S$ it is a trivial requirement.

$S$ may then proceed to compute the result of $*$ using his digital computer. E herself is presumed to be doing the same computation on her own computer. Now assuming the above and 1 in particular, $S$ knows that $*$ cannot halt or rather he knows that E must believe it cannot halt. For otherwise if $*$ halts with an integer answer $x$, instead of answering $x$, $S$ may answer $x+2$, (or something equally contradictory) and so he obtains an immediate contradiction, E is expecting $x$. On the other hand if $*$ does not halt with an integer answer for E, (which from a Turing machine point of view is possible, $S'$ is not a priori programmed to give an integer, it is only "asked" to do so via some input, which we don't know how $S'$ will interpret) he may just answer 8, again giving a contradiction, as E is expecting $S$ to answer exactly like $*$. So $*$ cannot halt at all.

The truth of this non-halting (given the validity of E's conviction in 1) is apparent to us human beings, and so to $S$ by assumptions (and so to E). Thus upon this reflection, $S$ first checks after whatever time that seems reasonable to him that his computer computation of $*$ has not halted. If it did he answers as above. That is obtain the value $x$ of the result of $*$ and answer E: $x+2$.

If $*$ did not halt $S$ may simply answer 8 (or anything else). He thus halts with an answer in any case, obtaining a contradiction, whether $*$ halts or not, at least if $*$ halted before $S$ answered. Now the reader may object: well perhaps the time $S$ chose to wait is too short, 8 is what was expected by E after all, and sometime after $S$ answers E, $*$ halts with $x = 8$. But E controls the speed of the computation $*$, so if this computation really halts she could set it up so that whatever time $S$ chose to wait would be sufficient, that is $*$ halts in that time. That is she could run the experiment once to calculate the time $t_0$ $S$ takes to answer and then run it again adjusting the processing speed of the computer in $S$'s room, so that $*$ halts in time less then $t_0$. Since S has no way to check the processing speed he cannot "conspire" to always answer too early. (We don't necessarily mean consciously conspire.)

The only remaining possibilities is that either after waiting for $*$ to halt and obtaining the result $x$, $S$ answers $x$ anyway, or $S$ does not even start the computation $*$, or finally that $*$ does not halt and $S$ remains silent indefinitely. In the fist two cases either $S$ did not understand the instructions, or chose not to obtain a contradiction. Since it would be too ridiculous that this held for every $S$, let us reject this possibility for the moment.

The final case, while in a sense interesting is clearly incredible, we are to believe that some human beings magically become completely unresponsive (for all time) on certain simple verbal instructions. (More formally the responses are not unambiguously interpretable as strings in *Strings*). The only way this is even in principle possible is if $S$ first reads the source code of $S'$, then "decides" that $S'$ computes him, and then becomes unresponsive, but what magic compels $S$ to even read this source code (which may take years). If he does not read the source code then there is absolutely nothing special to $S$ about the string $\Sigma$, it is more or less the same as "how is the weather?" and so expecting $S$ to become unresponsive is unreasonable. In any case if we denote by $U$ the set of possible (in nature) subjects $S$ which fall into one of the above possibilities, then our "physical" hypothesis is this:

**Hypothesis 2.** *The complement of $U$ is non-empty.*

(Naturally this author would like to claim he is not in $U$, but there is a philosophical issue, how exactly does he prove it?)

Let us then summarize the above argument more formally. Let $\mathcal{HO} \subset \mathcal{O}$ denote the subset of operators corresponding to human beings as above. There exists (assuming our hypothesis) an $S \in \mathcal{HO}$, (which we described in detail above) with the following property. Given any Turing machine $S'$, $S$

accepts as input the instructions 1-4 above, (with respect to the chosen $S'$) which takes the form of a string $\Sigma = \Sigma_{S'}$. $S(\Sigma)$ is defined by the meta-algorithm $X$: wait for $t_0$ amount of time for $*S'(\Sigma)$ halt, if $*S'(\Sigma) \to x$ answer $y \neq x$. If $*S'(\Sigma)$ does not halt in time $t_0$ answer 8. Here time $t_0$ is as measured by the independent observer $E$ using some auxiliary independent (of the computational process $*S'(\Sigma)$) physical clock.

Then the following is satisfied after it has been arranged, without "knowledge by $S, S'$", by the independent observer E, who can control the speed of the simulation of $*S'(\Sigma)$, that if $*S'(\Sigma)$ halts, it halts in time less then $t_0$:

(1) $S(\Sigma)$ is defined.
(2) $*S'(\Sigma) \not\to S(\Sigma)$.

By the last formula we have: $S'$ does not compute $S$, since $S'$ was arbitrary $S$ is not Turing computable.

*Remark* 2. There is no problem at all with $t_0$ measured by $E$ being small. (In any sense.) For $S$ "halts" in time $t_0$, so if he is computed by $S'$, we must conclude that a computer simulation of $S'$ can be constructed which halts in time slightly less then $t_0$, since $S$ - a physical machine, does so. We leave discussion of practical concerns for the next section.

□

## 3. Reinterpretation as an actual experiment

Given Hypothesis 2 our argument via thought experiment is a proof. My opinion is that this hypothesis is just some trivial assumption about physical reality, (biological operation of human beings). It does not seem to be possible to immediately test this in some way, but we may however reinterpret our thought experiment as an actual experiment at least in principle, we now describe this. Moreover if Hypothesis 2 is false, then instead this experiment produces at least in principle a fail proof way of "crashing" (sane, where sane is as defined below) human beings (that is making them completely unresponsive for all time).

If human thought processes were entirely Turing computations, then it would be possible to reverse engineer any particular human $S$ to a modelling Turing machine $S'$ computing $S$. This may be a very difficult undertaking technologically, but in principle and in practice at some point in the future must be possible. (For our practical purpose.)

We shall use notation and ideas from the thought experiment above. Then our test is the following. We suppose that E has a human subject $S$ in isolation in a room, we suppose that $S$ is as in the setup of our thought experiment above. Let $S'_0$ be a Turing machine computing the operator $S : Strings \to Strings$, and as before $S'_0$ is simulated in $S$'s computer. Here by "computer" we mean something rather general: a physical machine simulating $S'$. In practice, particularly if quantum effects play a serious role in the human brain, this may need to be a quantum Turing machine of some sort, otherwise there may be obstructions to existence of a sufficiently fast simulation for the purpose of the experiment, cf. Remark 2. Of course we may always simulate via a classical computer but this may be completely impractical.

Now E passes $S$ the instructions 1-4 from our thought experiment, but to make things simpler from testing point of view, we replace instruction 3 with: "follow the meta algorithm $X$ (consistently if asked to repeat) to give the answer to instruction 4, where $X$ is the meta-algorithm we described in our thought experiment with $t_0 = 1$ sec." Again for testing purposes E may in addition tell $S$ that if he follows the meta-algorithm $X$ he will receive some incentive. Of course these new 4 instructions represented by string $\Sigma = \Sigma_{S'_0}$ could just as well have been used originally, *without* change to the meta-algorithm $X$ of our thought experiment.

**Possibility 1.** Our subject $S$ answers, but always answers the same as $S'_0(\Sigma)$. As we explained in our thought experiment if $S$ answers then $E$ can eventually arrange that $*S'_0(\Sigma)$ halts before $S$ answers. Explicitly: after obtaining the reference time $t_0$ that $S$ takes to answer, E asks $S$ to repeat, while speeding up the computation simulating $*S'(\Sigma)$, without knowledge of $S$. We won't specify how (perhaps she can just completely replace the underlying hardware without knowledge of $S$). Since $S$

by assumption does not know how fast $*S'(\Sigma)$ runs, he cannot always (for all $S'$) answer before $*S'(\Sigma)$ halts, even if he changes his time to answer, which was $t_0$ before.

So we eventually get a very absurd situation where $S$ cannot for some reason ever contradict $S'(\Sigma)$, (give a different value). As already discussed we should probably reject this as a possibility at all. Or let us define $S$ to be *sane* if this does not happen, that is if eventually $S$ contradicts $S'(\Sigma)$.

**Possibility 2.** $*S'(\Sigma)$ does not halt, and our subject $S$ becomes unresponsive, unable to ever answer. This is a much more interesting possibility, (more conceivable). It is still incredible however for reasons explained in our main argument preceeding Hypothesis 2. In this case we have unfortunately done something "inhuman" we just "crashed" our $S$!

**Possibility 3.** The only remaining (and truly plausible) possibility is that $S$ responds but $S(\Sigma_{S'}) \neq S'(\Sigma_{S'})$ or $*S'(\Sigma_{S'})$ does not halt, and so $S'$ does not compute $S$ after all, a contradiction.

## 4. Some possible questions

*Question* 2. Does the same argument not also prove that S is not even deterministic?

By a deterministic machine, in non-specific terms, we mean here a machine that accepts certain inputs and *not necessarily computably* deterministically produces outputs. (Or deterministically produces a probability distribution for the output, recall that *we* also call this deterministic). Then the answer is no. If we try to run the same argument but with Turing machine replaced by a deterministic machine as above, we shall arrive at the following point. Suppose E has detailed information about S's working as a deterministic machine. Suppose she somehow passed all this information to $S$, in the form of some finite data. But even if she is able to determine $S$'s supposed output $S(\Sigma)$, $S$ may not be able to determine it himself, as it is no longer a matter of a computation. (Nor is it certain she can determine it herself.) Thus $S$ cannot contradict E's expectation since he is unable to determine what she expects, even though he can certainly answer using his natural faculties! Just to give a colorful example, the answer she expects from $S$ may depend on whether some particular 4 manifolds are diffeomorphic. The latter is a computationally unsolvable problem. There are also other difficulties with running our argument completely in this case, recall that an important point for us was E being able to speed up a certain calculation. This no longer makes any sense.

*Question* 3. Ok, but what if $S$ is a Turing machine producing probabilistic answers, that is the answer expected by E is given by a probability distribution?

It is a mostly trivial complication, the probability distribution is computable by assumption, so E needs only to keep repeating the same experiment, the same argument as before would invalidate that $S$ is a Turing machine to any requisite certainty.

## 5. Conclusion

We may conclude from above the following. Either there must be Turing non-computable processes in nature, and moreover they appear in the cognitive functioning of the human brain, or it is possible via simple explicit verbal input to make totally unresponsive for all time (in the sense discussed) some (we called them "sane") human beings. Strictly speaking other then the explicit verbal input we need to make available to these human subjects certain data, which may not be simple, but is in principle constructible. Nevertheless, amusing as the latter possibility is we should probably reject it.

As mentioned in the introduction although it is possible that Turing computable artificial intelligence will start passing Turing tests, given the former alternative it will always be possible to distinguish some human beings (not in $U$) from any particular modelling Turing machine. Human beings may not be in any way superior to Turing machines, (at least our argument does not immediately have such a conclusion) but they are certainly different. This of course still leaves the door open for non Turing computable artificial intelligence. But to get there we likely have to better understand what exactly is happening in the human brain.

## 6. Acknowledgements

Dennis Sullivan, Dusa McDuff and Bernardo Ameneyro Rodriguez for helpful discussions.

## References

[1] A.M. Turing, *On computable numbers, with an application to the entscheidungsproblem* , Proceedings of the London mathematical society, s2-42 (1937).
[2] ———, *Computing machines and intelligence*, Mind, 49 (1950), pp. 433–460.
[3] S. Bringsjord and H. Xiao, *A refutation of Penrose's Gödelian case against artificial intelligence*, Journal of Experimental & Theoretical Artificial Intelligence, 12 (2000), pp. 307–329.
[4] S. Daegene, *Non-computability of consciousness*, arXiv:0705.1617.
[5] S. Hameroff and R. Penrose, *Consciousness in the universe: A review of the orch or theory*, Physics of Life Reviews, 11 (2014), pp. 39 – 78.
[6] J.R. Lucas, *Minds machines and Goedel*, Philosophy, 36 (1961).
[7] K. Gödel, *Collected Works III (ed. S. Feferman)*, (1995).
[8] R. Penrose, *Emperor's new mind*, (1989).

*E-mail address*: yasha.savelyev@gmail.com

University of Colima, CUICBAS