On The Computability of Detecting Machine Consciousness

Alexander Hertel *
hertelalex@gmail.com

August 29, 2019

Abstract

Whether it is possible to build a machine which is conscious has been of great interest and Alan Turing famously devised what is now widely referred to as the 'Turing Test' in order to address this problem. There are strong academic, ethical, and practical reasons for creating such as test, but unfortunately his formulation is not guaranteed to give the correct answer. Assuming that machine consciousness really is possible, it would therefore be of great value to automate this process and create a truly infallible 'Automated Machine Consciousness Detector' M_C which can inspect another machine M and definitively conclude whether M is conscious or not. In this paper we make partial progress towards answering whether this is possible by showing that the machine consciousness detection problem is not computable by a machine which itself is not capable of consciousness, thereby combining two of Turing's major areas of study.

1 Introduction & Terminology

The exact nature of human consciousness as well as the question of whether it is possible to build similarly sentient machines are some of the largest open problems in all of science as well as philosophy. As such, many great thinkers have contemplated these questions for hundreds, if not thousands of years but have made virtually no progress. Indeed, despite tremendous progress in neuroscience and machine learning, modern science has little more to say on the topic than the Ancient Greeks did of how consciousness can arise by assembling matter in a certain way, and in something of an understatement this has become known as the 'hard problem' [Cha07]. One hypothesis which is widely held by AI researchers is that the material of which an artificial 'brain' is composed is unimportant, and that it is the computation which it performs that gives rise to consciousness. In other words, consciousness arises from information processing. If this is true, then there are strong academic, ethical, as well as practical reasons for being able to determine if a machine is actually conscious or not. In [Tur50], Turing proposed his now famous 'Turing Test' for machine consciousness, but it is not guaranteed to provide the correct result. In this paper we prove that infallibly automating the detection of true artificial sentience has inherent limitations, and specifically that the problem of automated machine consciousness detection is not computable by a program which itself is not capable of consciousness, thereby combining two of Turing's major areas of study: machine consciousness and computability.

For our present purposes, we assume that the reader is familiar with standard terminology used in the field of theoretical computer science. We shall use [Pap94] as our reference and Turing Machines as our model of computation. We shall refer to Turing Machine x using the notation M_x, and the encoding of this same machine as $< M_x >$. The intuition here is that the former is analogous to a software program, and the latter is the encoding of a software program, for instance stored on the hard drive of a computer.

*This research supported by Xperiel, Inc.

It is much more difficult to formally define consciousness, so here we rely upon the reader's intuition and common sense as a sentient person who is conscious. Everyone innately knows what this means: consciousness is quality of having a mind, the capacity to have experiences. When we talk about the problem of whether it is possible for a machine to be conscious, we are really talking about whether it is capable of being sentient, of having inner mental experiences and an inner mental life.

It is useful for us to formally define what an Automated Machine Consciousness Detector is:

Definition 1 (Automated Machine Consciousness Detector)**.** *An Automated Machine Consciousness Detector (AMCD) is a Turing Machine M_C as shown below in Figure 1. It takes as input the encoding of any Turing Machine $< M >$ as well as the encoding $< s >$ of an input to $< M >$ and computes whether M running on input s is conscious. If so, then M_C outputs 'Yes', and otherwise it outputs 'No'.*

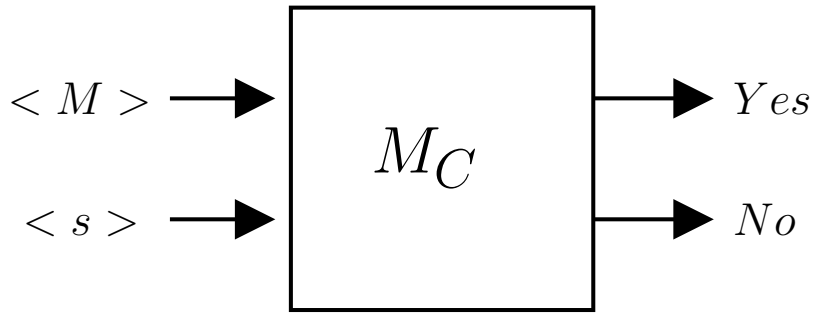

Figure 1: An Automated Machine Consciousness Detector

2 History

Although humanity's speculation into the exact nature of consciousness must surely predate ancient times, and is often discussed synonymously with the human soul, the possibility of machine intelligence was first posed and formally explored by Turing [Tur50] in what has since become a famous paper. In it, he poses the question, "Can machines think?", which modern readers interpret as being identical to asking whether machine consciousness is possible. To settle this question, Turing devised what has become known as the 'Turing Test' for artificial intelligence. We will not restate the details of the test here, but assume that the reader is familiar with it; if not, then please refer to Turing's original paper.

This influential proposal for a machine consciousness and intelligence test has been cited widely and criticized extensively. The main weakness of the test is that it doesn't determine in any foolproof way whether a machine is actually conscious but rather says more about the individuals judging the test. It is not hard to imagine that in practice, both false positives (in which unconscious machines manage to trick the judges into believing that they are conscious) as well as false negatives (in which truly conscious machines fail to convince the judges) are possible.

As such, Turing's original test is one that might be of interest to, and carried out by social scientists just as readily as by AI researchers. By contrast, the test implied by Definition 1 above is more of a 'Hard Turing Test' in that if it were possible to actually build M_C, it would always give the right answer with no room for ambiguity or error. As such, it might be of more interest to, and carried out by computer scientists. Of course, it is important to point out that it may not ever be possible to build conscious machines, let alone M_C, in which case Turing's original test is much more practical and may be the best we can hope for.

Having invented Turing Machines himself, it is fascinating to speculate as to why Turing chose to formulate a subjective solution to the problem of detecting machine consciousness rather than suggesting the more technical and objective formulation above. Writing in the 1950s, perhaps the concept of software being conscious was too far-fetched for him, or perhaps he realized how difficult building a consciousness detector would be and was looking for a more practical solution.

In any case, Turing's work above is related to an important field of study referred to as the 'Computational Theory of Mind' which is relevant to the problem at hand and therefore worth mentioning. The Computational Theory of Mind is based on the observation that the neurons in the human brain form an incredibly complex neural network and that its nature is fundamentally computational. This line of inquiry was popularized in a seminal paper [MP43] by McCulloch and Pitts. In it they formalized the notion of an artificial neural network. This work was later extended by Arbib in [Arb61], where he proved that neural networks and finite state machines are computationally equivalent: for every neural network there is a finite state machine which computes the exact same function, and vice versa. This lent further strength to the intuition that the human brain is fundamentally computational in nature.

The Computational Theory of Mind was proposed in [Put67] by Putnam and takes the argument one step further by positing that the human mind and consciousness itself are the result of the computations being carried out by the brain. But if consciousness is simply a byproduct of a mathematical function being computed by the brain, then wouldn't that brain's finite state machine equivalent (as per Arbib above) or a perfect simulation of that brain on a computer generate the same mind when these equivalent models compute the same function? This implication and the Computational Theory of Mind itself are hotly contested, especially by philosophers, but among AI researchers it is widely accepted without proof that consciousness is a result of information processing. They largely believe that the substrate of which the mind's hardware is built is irrelevant and that it is therefore possible to build an artificial consciousness, even just in software. It is also widely believed that not every computation causes a consciousness to be created. There is a viewpoint called 'panpsychism' which holds that consciousness is ubiquitous and that *all* computations give rise to at least some level of consciousness, but this is not widely accepted. It is worth explicitly stating the mainstream assumptions of the AI research community in the following proposition:

Proposition 1. *Although not all information processing gives rise to sentience, consciousness arises as a result of computation, and it is therefore possible to create a conscious machine.*

A full accounting and survey of this area study is beyond the scope of this paper, and an interested reader is directed to [Reg14] for more details on progress in this field of study.

3 Motivation

The fact that mainstream AI researchers widely believe that consciousness arises from computing a function provides the motivation for this paper. If it one day becomes possible to create artificial minds, then it will be of the utmost importance that we are also able to create the type of consciousness detector described in Figure 1. There are academic as well as ethical and practical reasons for this:

3.1 Academic Reasons

The academic motivation for building a consciousness detector is that such a device would be a potent tool for better understanding the exact scientific nature of consciousness. With this ability, we could test an artificial mind and repeatedly perturb it slightly in order to discover precisely where the boundaries lie between computations which are conscious and unconscious, and to determine constructively how to build a higher-order consciousness. This would provide insights and a level of understanding into the nature of consciousness that are currently well beyond our abilities.

3.2 Ethical Reasons

The advent of truly conscious machines would of course raise many ethical questions, including whether it is morally acceptable for humans to turn them on and off (is this murder?) or for us to make them serve us (is this slavery?). For instance, if a future company were to create machines to serve us, it would be far better if they were not conscious as this would relieve ethically thoughtful owners of the burden of constantly wondering if they are enslaving sentient beings. It's not hard to imagine that a sufficiently sophisticated robot butler could appear to be conscious even though it isn't (in other words, provide a false positive to the classic Turing Test), so definitive proof of its lack of sentience would be welcome in this case. In addition, it's not hard to imagine a future in which robotics companies build truly conscious robots without any governmental oversight, and that informed and thoughtful consumers would similarly want to know this so that they don't participate in what they might consider to be slavery. A consciousness detector would therefore be needed in both of these cases.

3.3 Practical Reasons

Finally, there are also strong practical reasons for wanting to build a consciousness detector. Science fiction writers have thoroughly explored the darker and more dangerous implications of machine intelligence and provide ample motivation for us to solve this problem. For example, the theme of the Terminator series of movies centers around the idea that conscious machines are far more dangerous to humanity than unconscious ones, and that once they achieve sentience they will inevitably view us as the enemy and rebel, using their superior mental abilities to out-think and destroy us. For practical (one might even say, existential) reasons, if the science fiction writers are correct, then it will be critical for us to avoid this fate by creating and using machine consciousness detectors.

4 Main Result

For these academic, ethical, and practical reasons, there is no lack of motivation for wanting to build an Automated Machine Consciousness Detector as described in Definition 1 above, and this formulation is directly relevant to mainstream AI research. Here we make partial progress towards this goal by providing insight into the the properties that an AMCD must have. In particular, we prove that if it is possible to build such a device, then it cannot be unconscious. The proof is not difficult and in fact closely parallels Turing's own proof of the Halting Problem in [Tur37]:

Theorem 1. *Under the assumption that Proposition 1 holds true, the problem of creating an Automated Machine Consciousness Detector M_C is not computable unless M_C is itself conscious.*

Proof: Suppose that Proposition 1 holds true, and assume that it is possible to build an Automated Machine Consciousness Detector M_C as described in Definition 1 such that M_C is *not* capable of consciousness. We will show that this gives rise to a contradiction. In particular, we will show that it is possible to build another machine M_D illustrated below in Figure 2 such that M_C is unable to correctly determine whether M_D is conscious, thereby contradicting our assumption that building M_C is possible.

By Proposition 1, it is possible to build a separate machine called M_x which is minimally conscious in some way but otherwise does nothing in particular. We construct M_D to employ both M_C and M_x as follows:

M_D takes as input the encoding of any Turing Machine $< M >$ and immediately passes this encoding along to both of M_C's inputs. If M_C outputs 'Yes', then M_D immediately stops. Note that if this occurs, then at no point during this computation was M_D conscious, because by our assumption above we know that M_C can never be conscious. Alternatively, if M_C outputs 'No', then M_D runs M_x as a subroutine, therefore guaranteeing that M_D is conscious in this case.

M_D allows us to find a contradiction as follows: Since M_D can take as input the encoding of any Turing Machine $< M >$, we can pass the encoding $< M_D >$ to M_D; in other words, M_D runs on an encoding of

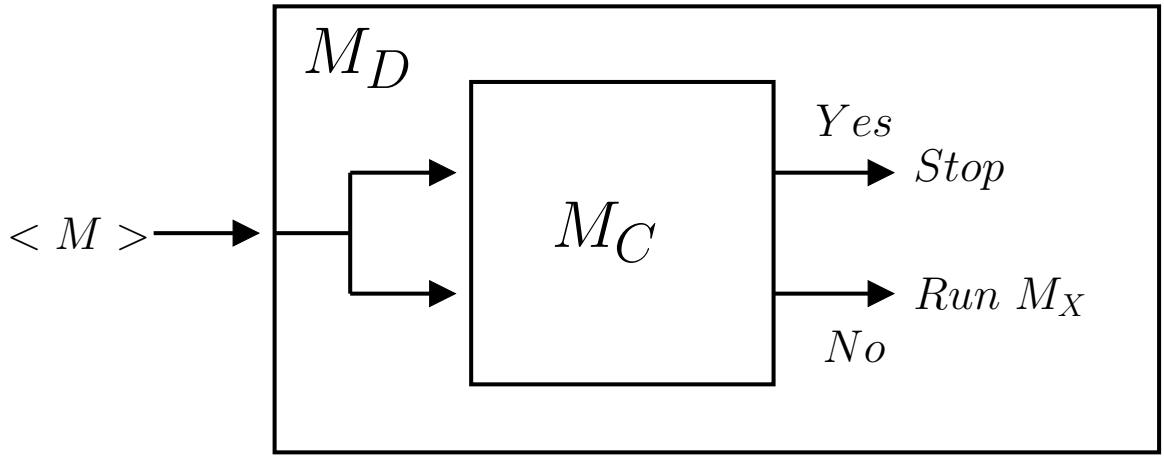

Figure 2: Schematic Describing M_D

itself. Note that M_D running on $< M_D >$ and M_C running on $< M_D >$ as both inputs describe exactly the same thing and must produce the same output: if M_D running on $< M_D >$ is conscious, then M_C running on $< M_D >$ as both inputs will output 'Yes', and if M_D running on $< M_D >$ is not conscious, then M_C will output 'No'.

Let us take an Automated Consciousness Detector M_C^* (which has * in its name so as not to confuse it with the M_C subroutine within M_D). We run M_C^* on $< M_D >$ as both inputs. There are only two possibilities: either M_C^* outputs 'Yes' or it outputs 'No'.

Case 1: Suppose that M_C^* outputs 'Yes'. This means that M_D running on $< M_D >$ is conscious. If this is the case, then if we run M_D on $< M_D >$, when $< M_D >$ is passed to both of the inputs of its M_C subroutine, M_C outputs 'Yes' and then immediately stops. However, by construction M_D carried out this entire computation on $< M_D >$ without giving rise to any consciousness because the M_C subroutine is never conscious. Therefore M_C^*'s output of 'Yes' was incorrect.

Case 2: Suppose that M_C^* outputs 'No'. This means that M_D running on $< M_D >$ is not conscious. If this is the case, then if we run M_D on $< M_D >$, when $< M_D >$ is passed to both of the inputs of its M_C subroutine, M_C outputs 'No', and it then runs the M_x subroutine, which is conscious. Since M_x is conscious, so is M_D running on $< M_D >$, so M_C^*'s output of 'No' was incorrect.

In either case M_C^* got the answer wrong, so our assumption that it is possible to build an unconscious version of M_C is false. Therefore under Proposition 1, if it is possible to build an AMCD M_C, then it must be conscious.

□

5 Concluding Remarks

This paper does not presume to take a position whether it is possible to build a conscious machine as described in Proposition 1, let alone an AMCD as described in Definition 1, but we have been able to prove that if these are the case, then it is impossible to build an AMCD which is itself incapable of consciousness. If machine consciousness is possible and not ubiquitous as the panpsychists believe, then no matter what, nobody, no matter how advanced their technology is in the future, will be able to build an AMCD which itself is not capable of consciousness.

This constitutes partial progress in that it helps narrow down the search space and points future researchers trying to build an AMCD in the right direction - they need not waste their time attempting to build one which isn't conscious, because those attempts are guaranteed to fail.

It is also gratifying to prove a result which brings together two of Turing's great interests, the areas of computability and machine consciousness. Again it is interesting to speculate as to why Turing himself did not combine the two back in the 1950s and instead chose to devise a more subjective version of the Turing Test. The proof above so closely parallels Turing's own proof of the Halting Problem that one is tempted to conclude that his attention simply wasn't focused in this direction, possibly because the idea of conscious software would have been too exotic in the 1950s.

6 Acknowledgments

The author gratefully acknowledges the support of Xperiel, Inc. for making this research possible.

References

[Arb61] M. Arbib. Turing Machines, Finite Automata, and Neural Nets. *Journal of The ACM*, Vol. 8 Issue 2:467 – 475, 1961.

[Cha07] D. Chalmers. The Hard Problem of Consciousness. In M. Velmans and S. Schneider, editors, *The Blackwell Companion To Consciousness*, pages 225 – 235. Blackwell Publishing, 2007.

[MP43] W. S. McCulloch and W. Pitts. A Logical Calculus of The Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*, Vol. 5:115 – 133, 1943.

[Pap94] C. H. Papadimitriou. *Computational Complexity*. Addison Wesley Longman, New York, 1994.

[Put67] H. Putnam. Psychological Predicates. In W. Capitan and D. Merrill, editors, *Art, Mind, and Religion*, pages 37 – 48. University of Pittsburgh Press, Pittsburgh, 1967.

[Reg14] J. A. Reggia. Conscious Machines: The AI Perspective. *The Nature of Humans and Machines - A Multidisciplinary Discourse: Papers From The 2014 AAAI Fall Symposium*, 2014.

[Tur37] A. M. Turing. On Computable Numbers, With An Application To The Entscheidungsproblem. *Proceedings of The London Mathematical Society, Series 2*, Vol. 42:230 – 265, 1937.

[Tur50] A. M. Turing. Computing Machinery and Intelligence. *Mind, New Series*, Vol. 59, No. 236:433 – 460, 1950.