On The Computability of Detecting Machine Consciousness

Alexander Hertel *
ahertel@cs.toronto.edu / hertelalex@gmail.com

December 29, 2023

Abstract

Whether machines can be conscious is 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 valuable to create a stronger and more fundamental formulation of Turing's Test and automate this process by building a truly infallible 'Automated Machine Consciousness Detector' M_C that can inspect another machine M and definitively conclude whether M is conscious or not. In this paper we make progress towards answering whether this is possible by proving three results. To date, formal mathematical proofs regarding machine consciousness have been hindered by researchers' inability to provide a technical definition of this phenomenon. In the present paper, we solve this problem by applying a proof technique that is more powerful than it needs to be and therefore holds for all reasonable definitions of consciousness. Our first result shows that the machine consciousness detection problem is not computable by any Turing Machine that is itself not conscious. The second result shows that under basic assumptions, a restricted form of the machine consciousness detection problem is computable. Finally, our third result builds upon the first two by showing that this restricted form of the problem again becomes incomputable if we limit the execution time of M_C by even a small fraction. These results unify two of Turing's major areas of interest: computability and AI. They highlight several boundaries relevant to machine consciousness detection, thus providing some new insights into what has been called the 'Hard Problem of Consciousness', and suggesting that the objective detection of machine consciousness may never be possible.

1 Introduction

The exact nature of human consciousness as well as the question of whether it is possible to build machines with this capacity 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 strides in neuroscience and machine learning, modern science has little more to say on the topic of how consciousness can arise by assembling matter than the Ancient Greeks did. The first known historical statement of this major question was by Democritus (c. 460-370 BCE) [Tay99] and traditionally this has been called the 'mind-body problem', but has since become known popularly as the 'Hard Problem of Consciousness', a moniker coined by Chalmers [Cha07].

One hypothesis called the 'Computational Theory of Mind' is widely held by AI researchers and states that the substrate material of which an artificial 'brain' is composed is unimportant, and that it is the computation it performs which gives rise to consciousness. In other words, consciousness arises from information processing. A more thorough history of this line of research is provided below in Section 2, and if this is true, then it implies that artificial minds are possible (even just in software), which

*This research made possible by Intuit, Inc.

immediately motivates strong academic, ethical, as well as practical reasons for being able to determine if a machine is actually conscious or not. These motivations are discussed below in Section 2.5.

In [Tur50], Turing proposed his now famous 'Turing Test' for machine consciousness, but it is not guaranteed to provide the correct result. Since then, many other tests for machine consciousness have been proposed [Hai19, p.194], but the goal of proving any formal results has been hampered by researchers' inability to provide a sufficiently rigorous definition of consciousness - after all, how can one prove exact mathematical results based on a squishy definition?

In this paper we present a novel solution to this problem: In Section 3 we review the terminology upon which our main results will rely, including an approximate definition of consciousness. We are able to solve the problems caused by the informality and insufficiency of this definition by applying a technique inspired by Rice's Theorem. In Section 4 we use this classic idea in a new way to prove results that are stronger and more general than they need to be, and therefore hold for *all* reasonable definitions of consciousness.

Our first result (please see Theorem 4.2) shows that infallibly automating the detection of artificial consciousness has inherent limitations, and specifically that the problem of automated machine consciousness detection is not computable by a program that itself is not capable of consciousness. This is followed by both positive and negative results regarding time-bounded versions of this problem (please see Theorems 4.3 and 4.4). These results definitively rule out specific possibilities and provide formal progress in an area of research where mathematical results have been difficult to achieve. More fundamentally they place limitations on our understanding of machine consciousness by suggesting that we may never be able to predict or detect it, and if we cannot do that, then we also cannot claim to have truly mastered or understood it. In addition, these results unify two of Turing's major areas of study: machine consciousness and computability.

Finally, in Section 5 we provide concluding remarks, and Section 6 outlines conjectures and future research related to our main results.

2 History

Humanity's speculation into the exact nature of consciousness must surely predate ancient times, and is often discussed synonymously with the human soul. Theologians, philosophers, and scientists have dedicated countless hours to pondering the problem of how it is possible for matter to be arranged in a way that hosts a mind, and more on the what this means is means can be found below in Section 3. In the present section we will review the academic literature relevant to this paper:

2.1 The Turing Test

The possibility of computationally-based machine intelligence was first formally explored by Turing [Tur50] in what has since become a famous paper. In it, he poses the question Q: "Can machines think?", which modern readers interpret as being essentially the same as asking whether machine consciousness is possible [Hai19]. After all, 'thinking' and 'computing' have different connotations, thinking being an anthropomorphization that implies consciousness. Instead of answering question Q which was too difficult, Turing reframed the problem to a more tractable form Q', consequently devising what has become known as the 'Turing Test' for artificial intelligence which asks whether a machine can pass this 'imitation game'. We will not restate the details of the test here, and assume that it is familiar to the reader.

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 determines whether the individuals judging the test believe that the machine *appears* to be conscious. It is not hard to imagine that in practice, both false positives (in which unconscious machines or algorithms such as ChatGPT 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 in the same way that a cat would fail the Turing Test, even though it is conscious) are likely. Incidentally, machines which exhibit sophisticated, human-like behavior without being conscious are referred to in more recent literature as 'zombies'.

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 because it says something about people. However, Turing did not need to reformulate the problem to Q', and in the present paper we deal directly with the original more fundamental question and propose the creation of an '*Objective Turing Test*' that addresses the original question Q by exploring the possibility of creating an 'Automated Machine Consciousness Detector' or $AMCD$ (described more formally in Definition 3.4 below) that can determine whether an algorithm is conscious when running on a given input. If it were possible to actually build such an $AMCD$, then 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 because it mainly tells us something about machines. Of course, it is important to point out that it may not ever be possible to build conscious machines, let alone an $AMCD$, in which case something fundamentally similar to 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 described below. One can only speculate: writing in the 1950s, perhaps the concept of software being conscious was too far-fetched for him because the computers of the day weren't powerful enough, or perhaps he struggled with a formal definition of consciousness, or realized how difficult building a consciousness detector would be and was looking for a more feasible solution.

2.2 The Computational Theory of Mind

In any case, Turing's work above is related to the important field of study called the 'Computational Theory of Mind' which is relevant to the problem at hand and therefore worth summarizing. The Computational Theory of Mind is based on the observation that the roughly 86 billion neurons in the average 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 that 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 built on these results and was proposed in [Put67] by Putnam. It takes the argument one step further by hypothesizing 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? The Computational Theory of Mind suggests that such software simulations would themselves be conscious. Put another way, the Computational Theory of Mind is closed under simulation because the simulation is computing the same function.

This implication and the Computational Theory of Mind itself are hotly contested, especially by philosophers. Indeed, because of a Turing Machine's step-by-step nature, it is difficult to see how it could possibly be conscious or cause consciousness to arise. After all, if there is no consciousness associated with it on step k of its computation, but on step $k+1$ there is, then what precisely could have happened between those steps to make it appear? Nevertheless the consensus among AI researchers is that machine consciousness is possible, and if that is the case, then Turing Machine consciousness must also be possible. Computers and software in the real world manipulate symbols in the same step-by-step manner as Turing Machines, so it would be hard to justify a belief that machine consciousness is possible, while

simultaneously denying this possibility of Turing Machines - after all, TMs are our foundational model of computation according to the Church-Turing Thesis. The Computational Theory of Mind is similarly contested by another competing theory of consciousness called 'Integrated Information Theory', or IIT which argues that in the same way that data center floors don't get wet when their computers run climate models, so too is consciousness a property of a physical system and therefore cannot exist purely in software [TK15].

2.3 AI Researcher Consensus

Nevertheless, among AI researchers the Computational Theory of Mind is widely accepted without proof and that consciousness is a result of information processing. They 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. Although it is difficult to find a comprehensive survey, a 2018 summary of interviews of 33 AI researchers revealed that only one member of this group believed that machines will never gain consciousness [Fag18].

It is also widely believed that not every computation causes a consciousness to be created. The other two possibilities aren't nearly as interesting from the point of view of detecting machine consciousness: There is a viewpoint called 'panpsychism' which holds that consciousness is ubiquitous and that *all* computations (or any physical interactions, for that matter) give rise to at least some level of consciousness. Despite having received some support from AI researchers such as McCarthy who in [McC79] proposed that even thermostats can be said to have beliefs, this is not widely accepted - instead, most AI researchers (as well as philosophers) hold that consciousness is a binary quality which is either present or not. At the opposite end of the spectrum, there are those who for various reasons (including religious ones) believe that it is impossible for a machine to be conscious (because consciousness is often equated with a human soul). In both of these cases, machine consciousness detection is trivial - it is respectively either all or nothing, and therefore also trivially computable - if the panpsychists are right, simply always output 'Yes', and if the machine consciousness deniers are right, then simply always output 'No'. The most interesting case is therefore one in which machine consciousness is possible but not ubiquitous, and this is the case that we will assume to be true for the purposes of this paper. This is also the position held by mainstream AI researchers and we will state this explicitly as Assumption 3.1 in Section 3 below.

A full accounting and survey of this area of study is beyond the scope of this paper, and an interested reader is directed to [Reg14] for more details on progress in this field. Further nuance introduced by quantum computers is discussed by Aaronson in his essay [Aar13], and a model for conscious Turing Machines that is very different from the one in the present paper is described in [BB22].

2.4 Incomputability & Rice's Theorem

As a final note to help place the present paper into the larger historical research context, the exploration of what is computable was also initiated by Turing himself in [Tur37], and this domain has been well-studied. It is widely known within the folklore of theoretical computer science that detecting any non-trivial property of a Turing Machine is incomputable, a fact proven by Rice [Ric53]. Rice's Theorem informs our intuitions and the incomputability of detecting any aspect of machine consciousness should therefore come as no surprise to anyone familiar with the literature. Rice's Theorem is relevant to the present result but as we'll see in Section 4.1 below, it cannot be applied directly and we will have to do some work ourselves.

2.5 Motivation

The motivations behind this area of study cannot be understated. The nature of consciousness is quite possibly the largest open problem in both science as well as philosophy, and therefore its study requires little justification. If, as is the goal of many major tech companies and startups as well as many academic researchers, it one day becomes possible to create artificial minds, then it will be of the

utmost importance that we are also able to detect which machines are conscious and which ones are not. Indeed, the prominent rise of large language models such as ChatGPT in 2022 - 23 reinvigorated and popularized the age-old question of whether machines can be conscious. There are ample academic, ethical, and practical motivations for studying this question:

2.5.1 Academic Motivation

The academic motivation behind 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 by repeatedly perturbing it slightly in order to discover precisely where the boundaries between minimally conscious and unconscious computations lie, 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 to reach.

2.5.2 Ethical Motivation

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 corporation were to create AI-powered products or machines to serve us, it would be desirable for them to be able to definitely prove that they are not conscious as this would avoid potential legal problems for the corporation and also relieve ethically-concerned owners of the burden of constantly wondering if they are enslaving sentient beings. It is not hard to imagine that sufficiently sophisticated AIs (including large language models) could appear to be conscious even though they aren't (in other words, provide a false positive to the classic Turing Test), so definitive proof of their lack of sentience would be welcome in this case. In addition, it is 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.

2.5.3 Practical Motivation

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 as an existential threat to human civilization 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 quickly improve themselves (also known as an 'AI Singularity'), and then out-think and destroy us. For practical and existential reasons, if the science fiction writers are correct, then it will be critical for us to avoid this fate and creating and using machine consciousness detectors would be invaluably helpful in discovering these AI threats. While these worries have clearly been dramatized, the fundamental underlying concerns are sound. Chaotic effects rising from complex systems notwithstanding, unconscious machines simply do what they are made to do. By adding the extra dimension of consciousness to them, they at least in principle gain the added motivation to harm us, and therefore are strictly more dangerous to humanity than their unconscious counterparts.

3 Terminology & Definitions

This section contains the terminology and definitions on which our main results will rest, along with discussion describing them. First we will describe our more technical terminology. This is followed by

the description, assumptions, and definition of what we mean by machine consciousness, and finally we will provide the definition of a formal Automated Machine Consciousness Detector.

3.1 Technical Terminology & Models of Computation

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 [Sip13] as our reference for computer science terminology. Turing Machines are our model of computation, and we will refer to Turing Machine X using the notation M_X, while the encoding of this same machine is $< M_X >$. 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 as an executable ASCII file stored on the hard drive of a computer.

It may seem odd to choose Turing Machines as our model for exploring conscious computations because they are so different from the human nervous system. Our brains are not finite state machines, and whatever the physical nature of our consciousness may be, we can be certain that it doesn't involve a Turing Machine's infinite tape in any literal sense. The way in which a Turing Machine accepts or rejects its input also doesn't seem analogous to how we function. Our nervous systems turn on before we're ever even born, and then receive a steady stream of inputs from our sensory neurons and similarly send a steady stream of outputs to our motor neurons without ever accepting, rejecting, or halting until the day we die. As such, one might argue that Finite State Transducers (FSTs) or something similar would be a more natural model of computation for us to choose if we want to at least get this aspect right because FSTs have two tapes that more naturally reflect sensory and effector neurons - one for inputs, and the other for outputs, and the output is a string rather than an accept or reject state.

Nevertheless, this paper assumes that the Computational Theory of Mind is correct, and it is agnostic to the model of computation being used or the substrate in which the relevant computation is implemented, be it a Turing Machine, a Finite State Transducer, or the biology of our brains. Turing Machines are able to simulate all other models of computation, and the Computational Theory of Mind is closed under simulation, so we can stick with convention and use the Turing Machine model with confidence that this doesn't affect the validity of our results.

3.2 Consciousness

It is much more difficult to formally define consciousness. Nobody has been able to provide anything resembling a technical definition, and to be clear, we won't be able to achieve this either. In line with the 'Hard Problem of Consciousness', we know so little about its physical nature that it seems unlikely one will soon be forthcoming. This lack of a formal definition has hindered researchers' ability to prove mathematical results regarding machine consciousness, so to solve this problem, our main results employ a proof technique inspired by Rice's Theorem that is more powerful and general than it needs to be. This makes our proofs insensitive to the specific nature of consciousness and allows them to hold true for all reasonable definitions thereof.

Nevertheless, next we provide a discussion and approximate definition of what we mean by consciousness so that we can all be assured we are aiming at roughly the same target. We can start by pointing out that in the same way that it is easy for a fish to forget it is wet, so too is it easy for us to forget that we are conscious creatures. This is because literally every experience we have happens within the medium of consciousness, and it's all too easy to take this amazing phenomenon for granted. Put another way, consciousness is necessary for having experiences, and it is impossible to have them without it. The word 'experience' isn't unique in this way, and in fact much of human language is indelibly marked by the fact that it was created by conscious creatures. We have many words that lose their most important dimension of meaning in a Universe where consciousness doesn't exist. If a being is not conscious, then it cannot have a 'mind', have 'mental' states, be 'sentient' or 'sapient'. It cannot 'think', 'concentrate', or 'focus' its 'attention'. It cannot 'hope', 'imagine' or 'feel' any 'emotions' such as 'sadness', 'happiness', 'guilt', 'envy', 'frustration' or 'pain'. We even have different degrees for these, such as 'depression', 'overjoyed', 'exasperated', as well as many types of pain, which can be 'sharp', 'dull',

'throbbing', 'burning', or 'shooting', among others. Our 'senses' give us thousands more words that have a first-person subjective experiential dimension to them. Our sense of 'sight' is a constant stream of virtually infinite visual experiences. Even just the colors such as 'red', 'blue', and 'green' in all of their shades are experiential qualia. The same goes for our sense of 'taste' - we have the basics such as 'bitter', 'sour', 'salty', and 'sweet', but this can be extended by the specific tastes of thousands of foods such as 'chocolate', 'shrimp', 'chanterelle mushrooms', 'lime', 'sun-dried tomatoes', and every other flavor on the planet. We can similarly repeat this exercise for our senses of 'smell', 'hearing', 'touch', as well as our internal senses such as 'hunger' to add many more phenomena to our lexicon of words that have experiential connotations to them for humans. Our languages are steeped in this experiential dimension, and these are only accessible to conscious creatures. Consciousness is the necessary software-like medium that is required, in which any and all of these mentalistic experiences must occur. This is worth dwelling on. Researchers refer to these experiences as 'qualia' [Tye21], which we will use shortly and therefore define here:

Definition 3.1 (Qualia)**.** *Qualia (singular quale) are any first-person subjective experiences such as thoughts, emotions, feelings, and sensory perceptions that beings are capable of having.*

When we talk about the problem of whether it is possible for a machine to be conscious, we are really discussing whether it is capable of having an inner mental life and the capacity to feel *any* of these types of qualia - even just the smallest flicker of one. Sometimes described as the "ghost in the machine", a rigorous definition of consciousness has eluded the scientific community ironically in part because it is itself the prime quale, and words therefore seem to be inadequate for capturing it, much in the same way that no mere words seem to be sufficient for conveying the precise taste of a strawberry or the color yellow to anyone who hasn't respectively tasted one or seen it before. Some insight is offered by Haikonen in [Hai20] where he suggests that qualia are self-explanatory in an atomic sense - that is, anyone able to experience qualia already has something self-explanatory that is far better than a written definition and therefore doesn't need one. Ironically, the language of poetry rather than that of science can often be better at capturing these seemingly ineffable quantities. Similarly ironic is the observation that if it truly is possible to build conscious machines in software, then qualia can be captured alphabetically. This is because any future conscious software will be encoded as strings, which means that any qualia that they experience can also be encoded as strings. It is irritating to consider that we humans may always struggle to find the words to describe a specific taste of an apple to each other, whereas a conscious machine might simply be able to share the appropriate code with its machine friend, which then runs it and experiences the taste of an apple.

Another observation to be made is that if the human mind is indeed a type of computer, it seems to be very different from their digital counterparts in the sense that our computations appear to be entirely of a semantic nature - that is, when we humans "compute", this can occur within a medium of consciousness where everything is imbued with a sense of semantic meaning and experience. Even when performing computations such as arithmetic or deciding which chess piece to move, these syntactic computations by humans are, for the lack of a better term, being "simulated" in a meaning-laden experiential context of consciousness. This is perhaps an appropriate definition for the word 'think' - these are the computations that are being carried out and experienced within the medium of consciousness. As a concrete example, use your mind's eye (which your consciousness is 'seeing') to picture a black chalkboard with the equation $2 + 2 =?$ written on it in white chalk. Again in your imagination, picture yourself approaching the chalkboard, and replacing the ? with the number 4. You just carried out a computation in your consciousness. By contrast, as far as we can tell, our current digital computers and other devices are entirely limited to purely unconscious, syntactic computations and symbol manipulations. When a computer adds $2 + 2 = 4$, it doesn't imagine anything but rather does so by loading these numbers in binary form into registers and sending them to its arithmetic logic unit within its CPU which manipulates the bits in order to perform the addition. When you performed the calculation, you experienced it in consciousness. Very few people believe that the computer is experiencing anything when it calculates the exact same function. Similarly, when a robotic vacuum cleaner makes the relevant calculations to clean our floors, few people believe that the "lights of consciousness are on". Beyond calculations, and as

we've already discussed, the experience of sensations requires consciousness in which to feel the sensation, so to a robot with a temperature sensor, hot and cold are just different numbers, whereas to us they *feel* different, thus illustrating that something fundamentally different is going on. Nevertheless, human consciousness is software-like, and is even loosely analogous to a computer operating system, while qualia being experienced within the medium of consciousness are loosely analogous to computer programs being run by that operating system. Like software, it is reasonable to hypothesize that consciousness has no mass because as far as science can tell, there is neither a weight decrease in a CPU when it is rebooted or turned off, nor in a creature when it loses consciousness or dies.

There have been many insightful discussions on the topic and an interested reader is directed to [Cha96, Hai19, Har19] for thoughts beyond the scope of this paper.

One thing is for certain: we live in a Universe where the physical laws of nature allow consciousness to exist, and for our purposes it is therefore meaningful to focus on the concept of a machine having at least 'minimal consciousness'. To understand what we mean by this, it is helpful to look at the rest of the animal kingdom. Descartes' animal machine argument notwithstanding, most people today find it to be intuitively clear that humans aren't the only conscious creatures, and that it is possible for other life forms to be less conscious than we are while still having some level of consciousness. This is a well-studied domain of research and in 2012 a conference at Cambridge University brought together prominent experts in neuroscience and related fields. They produced the *Cambridge Declaration on Consciousness* [Low12] which states that a neocortex is not necessary for consciousness and that many "non-human animals, including all mammals and birds, and many other creatures, including octopuses" possess the necessary neurological structures for them to be conscious.

This is no surprise to virtually all dog or cat owner who would attest that their pets have personalities and are therefore conscious. However, this becomes less and less clear as we descend through the tree of simpler organisms. Are mice or birds conscious? The Cambridge group as well as most lay people would probably say yes. What about frogs? Fish? Spiders? Worms? Amoebas? Plants? Mushrooms? Bacteria? Viruses? Somewhere along this gradient, common intuition suggests that there is a cutoff point below which creatures lack the structures such as a nervous system necessary to support any consciousness whatsoever, and it is therefore widely believed to be binary (which coincides with the consensus among AI researchers regarding machine consciousness), but it is also a spectrum. In other words, there is a binary cutoff point below which there is no consciousness, and above which there is a spectrum. Somewhere near the bottom of the spectrum there might be a creature that can experience, say, the feeling of hot and cold, but not much else. We don't need to know exactly where this cutoff point is, and indeed our results hold regardless of where we find this threshold. Instead we will rely on the concept of a machine possessing at least some minimal level of consciousness that goes beyond mere calculation.

As described above in Section 2.3, the mainstream AI research community subscribes to the Computational Theory of Mind, which we assume to be true of the purposes of this paper:

Assumption 3.1. *The Computational Theory of Mind as interpreted by the mainstream AI research community is correct: Although not all information processing gives rise to consciousness, and it is a binary quality which is either present or not, it arises as a result of certain computations but not others, and it is therefore possible to create a conscious machine, even just in software. Furthermore, the consciousness of a computation is substrate-independent - if a computation gives rise to consciousness when carried out on one machine / medium / model, then it will be conscious when carried out on all others, provided that they are sufficiently powerful to carry out that computation. Computational consciousness is therefore also closed under simulation because the simulation of a conscious machine is computing the same function.*

This assumption together with the intuitions described in the paragraphs preceding it lead to the following definition of consciousness, which also captures the Computational Theory of Mind's closure under simulation as well as its concept that ordinary computations of the type we've seen in typical software programs such as web browsers, PDF viewers, and printer drivers are not conscious. One day

we may know enough about the physical laws underlying consciousness to provide more rigorous details, but Definition 3.2 below will suffice for our purposes because of the robustness of our subsequent proof techniques in Section 4, which are more general than they need to be. This makes our results insensitive to specific details regarding the definition of consciousness, allowing this approximate definition to be sufficient for our needs:

Definition 3.2 (Approximate Definition of Conscious Turing Machine)**.** *Consciousness is the massless, software-like medium in which qualia are experienced. More specifically for machines, let M be a Turing Machine running on input s. Then both M and its computation on s are conscious if and only if at any point during its computation, M experiences any qualia. In addition,*

- *The simulation of consciousness is conscious: If a Turing Machine M_1 running on input s is conscious, and if Turing Machine M_2 simulates the computation of M_1 on s, then this resulting computation by M_2 is also conscious, and*
- *Ordinary computations are not conscious: If a Turing Machine M_T encodes any algorithm of the type found in typical legacy software programs such as word processors, spreadsheets, video games, operating systems, etc. created by humans before the year 2020, then M_T is not conscious when run.*

There is some confusion about the nature of machine consciousness that Definition 3.2 helps to clarify. Often, when defining something, it can be helpful to provide adjacent counterexamples. For instance, there is a distinction to be made between machine consciousness and the notion of an artificial general intelligence, or AGI. These are not the same thing but are often conflated. Although it is widely assumed that an AGI would be conscious, some have argued that this isn't necessarily the case. However, even if it is, one does not have to try very hard to imagine a machine consciousness (for example the one described in Definition 4.1 below) that is not an AGI. This mirrors the situation with biological consciousness because humans are conscious and have general intelligence, whereas cats are conscious but do not. This shows that machine consciousness and AGI are not synonymous.

Further confusion is caused when sleep is conflated with unconsciousness, but we know that the human brain is capable of consciousness even when we are sleeping because we are able to dream, and those dreams are clearly being experienced by a consciousness. Patients in comas and under certain types of anesthesia have similarly reported dreams, illustrating that the term 'consciousness' is often used interchangeably with the state of being awake. This shows that the word has more than one meaning, but it is important to understand the distinction between them - unconscious as in not awake does not mean that consciousness is not present.

Yet another clarifying distinction has to do with the notion of self-awareness. Consciousness and self-awareness are often conflated, but again these are not the same thing. Humans are both conscious and self-aware, with a high capacity to reflect on our own consciousness. There are other animals that we believe to be conscious but lack this type of self-awareness - indeed, most of them are not even able to recognize themselves in a mirror. It is not hard to imagine that some lower life forms could have a very rudimentary consciousness that allows them to experience the feeling of, for example, the difference between hot and cold, but completely lack some of the higher-order conscious capabilities such as the self-awareness that we possess. On the other extreme, it is interesting to speculate whether there are conscious capabilities above us on this spectrum that humans lack, and what these might be.

This suggests a type of 'hierarchy of conscious capabilities', even within specific narrow domains. For example, evolution has imbued many creatures with the capability of feeling fear in order to avoid predators, and it is not unreasonable to believe that the visceral terror that many animals would feel if a lion jumped out of the bushes nearby is similar to what we would feel. However, there are many types of fear, and some forms of this very basic emotion that are available to humans cannot be experienced by other creatures. For instance, the fear of future abandonment is more of an abstract fear that is not situated in the present. Can a dog feel the fear of future abandonment? Are they even capable of modeling the future in this way? If so, then the point still stands - we need only descend through

the web of life. Can a mouse feel the fear of future abandonment? A bird, fish, grasshopper? As with consciousness, it seems likely that there is a threshold somewhere.

It is important to define what we mean when we say that a Turing Machine is capable of consciousenss. For each Turing Machine M there are three possibilities: 1) M is not conscious running on any inputs, 2) M is conscious when run on one or more inputs, but not on others, and 3) M is conscious running on all inputs. This allows us to define whether a Turing Machine is capable of consciousness:

Definition 3.3 (Turing Machine Capable of Consciousness)**.** *A Turing Machine M is said to be capable of consciousness if there exists an input s such that M is conscious when run on s.*

In practice, most people would not believe that the typical algorithms that they use in their day-to-day lives are capable of consciousness, regardless of whether they are implemented in software or hardware. If machine consciousness is possible, then our word processors are not likely to be the best place to go looking for it. By contrast, emulators built entirely in software are instantiations of Universal Turing Machines, so by Assumption 3.1, then in principle they are capable of consciousness when running on sufficiently powerful computers.

Other examples of non-conscious software are given in Definition 3.2, but it is equally unlikely that our hardware devices such as pocket calculators, television remote controls, and stereo systems are capable of consciousness, provided that they don't contain sufficiently powerful CPUs and sufficient memory. General-purpose computers, on the other hand, are also instantiations of Universal Turing Machines, so like the software emulators described above, if they have enough computational resources to, for example, run M_χ from Definition 4.1 below, then these hardware instances are in principle capable of consciousness.

3.3 Automated Machine Consciousness Detectors

Finally, it is important for us to define the nature and purpose of an Automated Machine Consciousness Detector ($AMCD$). If such an $AMCD$ were to exist, this would constitute an objective version of the Turing Test:

Definition 3.4 (Automated Machine Consciousness Detector)**.** *An Automated Machine Consciousness Detector is a Turing Machine M_C as shown below in Figure 3.1. 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 at any point during its computation. If so, then M_C outputs 'Yes', and otherwise it outputs 'No'.*

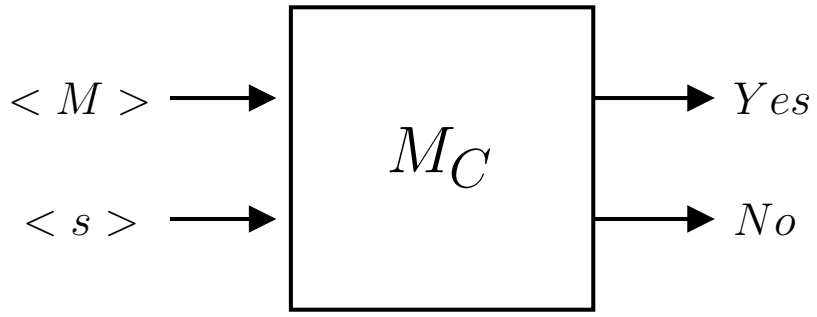

Figure 3.1: An Automated Machine Consciousness Detector

4 The Computability & Incomputability of Different AMCDs

For the academic, ethical, and practical reasons described in Section 2.5, there is no lack of motivation for wanting to build an $AMCD$ as described in Definition 3.4 above, and this formulation is directly relevant to mainstream AI research. Here we make progress towards this goal by proving three results that provide insight into the the properties that an $AMCD$ must have. Our first result shows that it is not possible to build an $AMCD$ that is itself not conscious. In other words, if it is possible to build such a device at all, then it must itself be conscious. Our second and third results respectively show computability and incomputability results for the more practical time-bounded form of this problem. Together these results let us map out some of the boundary conditions around the possibility of building machine consciousness detectors.

4.1 Inapplicability of Obvious Previous Techniques

At first glance to experts in theoretical computer science, the idea that Turing Machines can't determine if other Turing Machines are conscious may seem obvious for two reasons: Firstly, there appears to be a straightforward reduction from the Halting Problem to the problem of detecting machine consciousness. Secondly, this result also seems to follow directly as a corollary from Rice's Theorem [Ric53], which (informally) states that determining any non-trivial property P of a Turing Machine is not computable. In this Section we will gain some insights by showing how this obvious reduction fails, and also by showing that Rice's Theorem similarly can't be used as an easy shortcut to our result.

4.1.1 Failure of Obvious Reduction

The obvious reduction from the Halting Problem to prove Rice's Theorem fails as follows: Let us assume that there exists an $AMCD$ solver called M_C. The Halting Problem takes as input machine M and input s and we reduce it to M_C as follows: create a machine M^* that takes inputs $< M >$ and $< s >$ and creates $< M' >$ which itself takes no inputs but immediately runs M on s and only then runs some algorithm A that is known to be conscious but doesn't do anything else. M^* then passes $< M' >$ into M_C with nothing for M_C's secondary input and returns the same result that M_C returns. Within M^*, M_C reports that M' is conscious if and only if M' reached and ran algorithm A, but it could only have reached A if M had halted on s. M_C has therefore allowed us to build a functioning Halting Problem solver M^*, which is a contradiction.

Unfortunately this reduction doesn't work because we cannot assume that M_C's conclusion (that M' is conscious) came from A. It could be that M runs infinitely on s but does so consciously, in which case M_C gave us a false positive for halting by detecting unexpected consciousness in M running on s before it even got to A. (Note that M running on s can be conscious even if the panpsychists are wrong, and not every computation gives rise to consciousness.) This illustrates the difficulty at hand: we simply don't know enough about the nature of consciousness to easily adapt M so that it is guaranteed not to be conscious running on s, so a reduction in this manner appears to be very difficult to construct. Nevertheless, seeing exactly how this reduction fails does help to provide some insights into the nature of machine consciousness.

4.1.2 Inapplicability of Rice's Theorem

Originally published in 1953, Rice's Theorem is a broad and powerful result that can be formulated as follows [Sip97]:

Theorem 4.1 (Rice's Theorem)**.** *Let P be a property of Turing Machines and let $P_{TM} = \{< M > \mid M$ is a Turing Machine that has property $P\}$. If P_{TM} satisfies the following two conditions:*

1. *There exist Turing Machines M_1 and M_2, where $< M_1 > \in P_{TM}$ and $< M_2 > \notin P_{TM}$. In other words, P_{TM} is non-trivial - it holds for some, but not all Turing Machines.*

2. *For any Turing Machines M_1 and M_2, where $L(M_1) = L(M_2)$, we have $< M_1 > \in P_{TM}$ if and only if $< M_2 > \in P_{TM}$. In other words, the membership of a Turing Machine M in P_{TM} depends only on the language of M.*

then P_{TM} is undecidable.

One might think that machine consciousness detection is a special case of this result, and its non-computability should therefore follow trivially, but unfortunately this doesn't work. If we try to use Rice's Theorem to show that machine consciousness detection is not computable, here is what goes wrong: Let $P_C = \{< M > \mid M \textit{ is a Turing Machine that is capable of consciousness}\}$. Let M_1 be identical to M_χ from Definition 4.1 below, so M_1 is capable of consciousness, $< M_1 > \in P_C$, and $L(M_1)$ is the set of all strings. Let M_2 be the trivial Turing Machine that immediately accepts any input that it is given. By Definition 3.2, M_2 is not capable of consciousness on any input, so $< M_2 > \notin P_C$. P_C is therefore non-trivial and satisfies the first condition of Rice's Theorem. However, $L(M_2)$ is also the set of all strings, so $L(M_1) = L(M_2)$, but $< M_1 > \in P_C$ while $< M_2 > \notin P_C$, so P_C violates the second condition of Rice's Theorem, thereby showing that we cannot use it as an easy shortcut to prove that detecting machine consciouseness is not computable.

Had this worked, Rice's Theorem would have provided a stronger version of Theorem 4.2 which applies to all $AMCD$s, conscious or not, but instead we must settle for our weaker version. We will discuss the more general result as a conjecture in Section 6.

Nevertheless, even though we cannot apply Rice's Theorem directly, this exercise has given us some insights, namely that Rice's Theorem is fundamentally about the properties of languages decided by Turing Machines, whereas consciousness is not. Indeed, we could prepend M_χ as a subroutine to a Turing Machine that accepts any arbitrary language, thereby showing that machine consciousness can be completely divorced from the language it accepts.

In addition, we use Rice's Theorem to guide our intuitions and even find inspiration in its technique. In particular, it is extremely powerful, and we will use this overly-general sledgehammer in the form of a proof technique similar to those of Rice and Turing to overcome our inability to rigorously define consciousness above.

4.2 Incomputability of Non-Conscious AMCDs

Instead of proving our first result by reduction, we therefore opt for a more direct approach. The proof is not difficult and in fact closely parallels a direct proof of the Halting Problem, albeit in a more general manner similar to Rice's Theorem. We will make use of the following Subroutine:

Definition 4.1 (M_χ and χ)**.** *By Assumption 3.1, it is possible to build a Turing Machine that is minimally conscious but otherwise does nothing in particular. Let us define M_χ as the Turing Machine that ignores its input and requires the smallest number of computational steps χ in order to achieve consciousness and then halts and accepts.*

Intuitively, M_χ is one of the simplest possible conscious machines in that it is able to achieve consciousness in fewer steps than any other machine. As an aside, it is interesting to speculate as to what computational resources are required to run a real-world implementation of M_χ. Does this require a supercomputer of the future, or is an early smartphone sufficient? In any case, we will use this as a subroutine in a construction required by our main result:

Theorem 4.2. *Under the premise that Assumption 3.1 holds true, it is not possible to create an Automated Machine Consciousness Detector M_C that is itself not capable of consciousness.*

Proof: Suppose that Assumption 3.1 holds true, and assume that it is possible to build an $AMCD$ M_C as described in Definition 3.4 such that M_C is *not* capable of consciousness. We will show that this gives rise to a contradiction, in particular, that it is possible to build another machine M_D illustrated

below in Figure 4.1 such that M_C is unable to correctly determine whether M_D is conscious, thereby contradicting our assumption that building M_C is possible.

M_D makes use of M_χ from Definition 4.1 as a subroutine. By construction, M_χ is the only part of M_D that is capable of consciousness. We construct M_D to employ both M_C and M_χ 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, and by Definition 3.2, none of the additional mundane 'plumbing' in M_D is conscious. Alternatively, if M_C outputs 'No', then M_D runs M_χ as a subroutine, therefore guaranteeing that M_D is conscious in this case because by Definition 3.2, the simulation of a conscious machine is itself conscious.

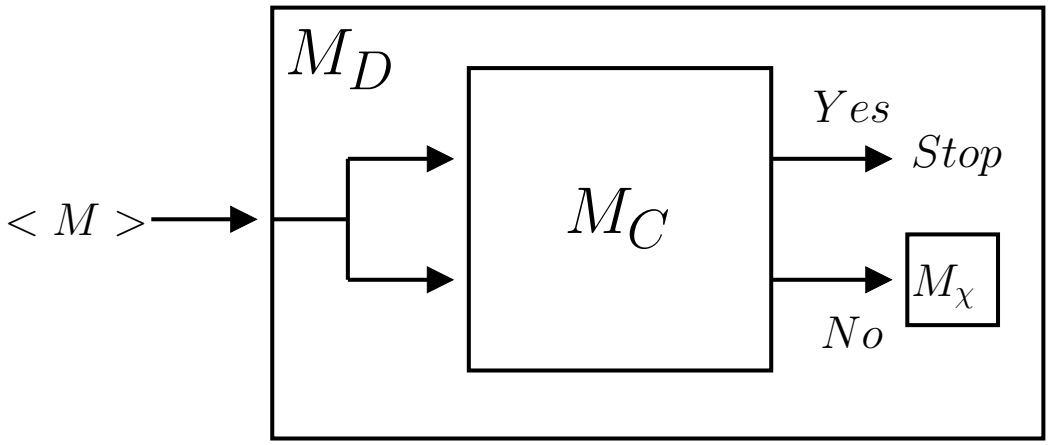

Figure 4.1: Schematic of M_D for Theorem 4.2

M_D gives rise to 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 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 should output 'Yes', and if M_D running on $< M_D >$ is not conscious, then M_C will output 'No', but we will see that they give opposite answers.

Let us take an $AMCD$ 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' (because it is the same as M_C^*) 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' (because it is the same as M_C^*), and it then runs the M_χ subroutine, which is conscious. Since M_χ is conscious, so is M_D running on $< M_D >$, so M_C^*'s output of 'No' was incorrect.

In both cases M_C^* produced the wrong answer, so our assumption that it is possible to build an unconscious version of M_C is false. Therefore under Assumption 3.1, if it is possible to build an $AMCD$ M_C, then it must be conscious, as required. □

The careful reader will note that with only slight modification to the proof of Theorem 4.2, one can substitute virtually any non-trivial property P of a Turing Machine instead of consciousness and it will still hold. At first glance, this may suggest that there is something wrong, but this same criticism can be leveled against Rice's Theorem, which we know to be perfectly correct, and have already discussed above in Section 4.1. This insensitivity to the particular details of P (in this case, consciousness) allows us to overcome the weakness and lack of rigor in Definition 3.2 above and holds for any non-trivial property P, let alone any reasonable definition of consciousness. Rice's Theorem already subsumes many such properties, but it is worth generalizing Theorem 4.2 to all qualia in the following Corollary:

Corollary 4.1. *If Assumption 3.1 holds true, then for all qualia Q it is not possible to create a Turing Machine M_Q that takes as input the encoding of another Turing Machine $< M >$ as well as the encoding $< s >$ of an input to $< M >$ and computes whether M running on input s experiences Q where M_Q is itself not capable of experiencing Q.*

4.3 Computability & Incomputability of Certain Time-Bounded AMCDs

Although the Halting Problem is undecidable, it is well-known that a special case of it is computable. The N-Step Halting Problem asks whether a machine $< M >$ running on input $< s >$ will halt within n steps, and the algorithm for solving this problem is straightforward: simply simulate $< M >$ on $< s >$ for n steps. If it halts within that time, then output 'Yes', and otherwise output 'No'. In this section we will examine the corresponding special case of the $AMCD$ problem and show one positive as well as one negative result. Under basic assumptions, we prove that this problem is computable. We then restrict this problem in a small but critical way and prove that this new version is not computable by an N-Step $AMCD$ which is itself not capable of consciousness.

To begin, let us formally define what we mean by this type of $AMCD$ which is time-bounded in the maximum number of steps for which its input is permitted to execute:

Definition 4.2 (N-Step Automated Machine Consciousness Detector)**.** *An N-Step Automated Machine Consciousness Detector is a Turing Machine M_C that takes as input the encoding of any Turing Machine $< M >$, as well as encoding $< s, n >$ of inputs to $< M >$ and computes whether M is conscious at any point when run on input s for n computational steps. If so, then M_C outputs 'Yes', and otherwise it outputs 'No'.*

Diagrammatically, our N-Step M_C is shown below in Figure 4.2. It is nearly identical to its counterpart shown above in Figure 3.1, with the exception that instead of taking $< s >$ as input, it takes $< s, n >$ as input.

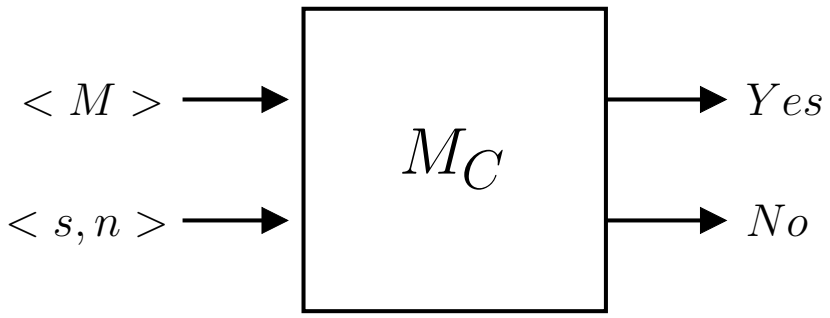

Figure 4.2: An N-Step Automated Machine Consciousness Detector

4.3.1 Computability of Time-Bounded Machine Consciousness Detectors

The N-Step $AMCD$ problem seems more practical than its unrestricted counterpart because it involves a fixed number of steps and thus avoids any issues of programs running infinitely. To show that this is computable, we will make use of two assumptions, both of which are basic claims about the nature of reality as well as the ability to simulate it. The first of these is essentially a statement of the same assumption that underpins all of science, namely that the the Universe is governed by laws of nature that describe all phenomena:

Assumption 4.1. *The phenomenon of consciousness in the physical world obeys some laws of physics, even if we don't currently understand what those laws are.*

It is difficult to imagine a rational argument denying this assumption because the alternative is that consciousness is supernatural and exists outside the laws of physics.

Our final assumption states that laws of physics that are known and understood can be simulated if given enough computational resources, and that this can be done in a manner such that the simulation can report which laws came into play during the simulation. For example, if it simulates a cannon shooting a cannonball, then it is able to accruately run the simulation but also report that the Law of Gravity participated in it. Similarly, if it simulates a consciousness, then it is able to report that the laws of physics governing consciousness participated in the simulation.

Assumption 4.2. *Given a sufficiently powerful computer C, the laws of physics can be simulated on it in a way that C can detect which laws were invoked.*

This allows us to prove the following result:

Theorem 4.3. *If Assumptions 4.1 and 4.2 are true, then it is possible to create an N-Step Automated Machine Consciousness Detector M_C.*

Proof: Suppose that Assumptions 4.1 and 4.2 are true. By these assumptions, it is possible to create a sufficiently powerful physical computer C shown below in Figure 4.3 that is capable of running a software simulator of the real laws of physics called M_C that takes as input the encoding of any arbitrary Turing Machine $< M >$, along with the encoding $< s, n >$ of an input to $< M >$.

Physics simulator M_C generates the software model P of a physical computer that takes as input the encoding of the Turing Machine $< M >$ as well as the encoding $< s, n >$. To be clear, M_C is not simply emulating a software version of P but rather is simulating a full physical model of computer P running $< M >$ on $< s >$ for n steps. Put another way, in this simulation, P is physical, whereas $< M >$ and $< s >$ are software. The indicated simulation is replicating what would happen if P were executing $< M >$ on $< s >$ in the real world.

By design, P runs $< M >$ on $< s >$ for exactly n steps. If, at any point during this simulation, P detects that the laws of physics governing consciousness have been invoked, it outputs 'Yes'. On the other hand, if it completes all n steps of simulation and never enacts any laws of physics governing consciousness, then it outputs 'No'. M_C is therefore not just a simulator of the laws of physics, but also comprises a correct N-Step Automated Machine Consciousness Detector M_C, thus showing that this problem is computable, as required. □

It is worth noting that according to Definition 3.2, this time-bounded N-Step $AMCD$ M_C is itself capable of consciousness because through simulation, it is computing the same function that $< M >$ would be computing on $< s, n >$, thus giving us the following corollary:

Corollary 4.2. *If Assumptions 3.1, 4.1, and 4.2 are true, then it is possible to create an N-Step Automated Machine Consciousness Detector M_C that is capable of consciousness.*

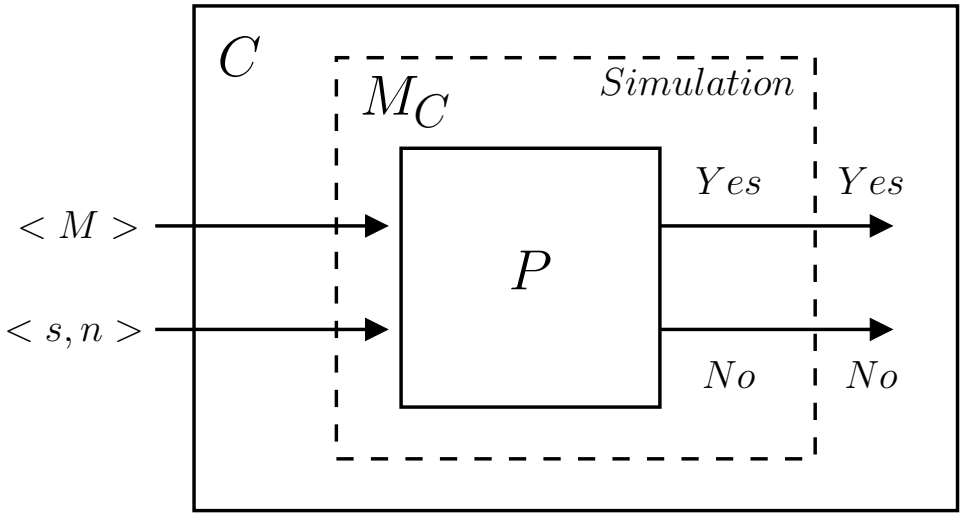

Figure 4.3: Schematic describing computer C running M_D, which is in turn simulating computer P running M on s for n steps

Unfortunately, in practical terms these results do not bring us much closer to actually building an N-Step $AMCD$ because realizing the assumptions upon which they are based would require us not only to solve the 'Hard Problem of Consciousness' but also to master computer simulation well beyond current human capabilities.

4.3.2 Incomputability of Time-Bounded Machine Consciousness Detectors

We will now make two modifications to the class of N-Step $AMCD$s and show that this makes it impossible to compute whether the machines described by their inputs are conscious. In Theorem 4.3, we created a simulator that simply runs M on s for all n steps, but what if we rule out the possibility of this type of simulation by stipulating that M_C cannot be conscious and must itself use fewer than n steps in order to compute whether M is conscious when run on s for n steps? In this section, we will show that restricting the number of steps that M_C is allowed to use to even just a small number below n while simultaneously restricting M_C to be incapable of consciousness once again makes this problem incomputable.

Our proof follows the same template as Theorem 4.2. We will again assume that it is possible to build such an N-Step $AMCD$ M_C, use it as a subroutine to build an M_D, and then use M_D to 'break' M_C. In this case, the construction of M_D is slightly more complicated than before, and is shown below in Figure 4.4.

The computation performed by M_D proceeds as follows: it takes as input the encoding $< M >$ and duplicates it. Unlike the M_C from Definition 3.1 whose second input is of the form $< s >$, our present M_C is expecting a second input of the form $< s, n >$. One duplicate of $< M >$ is sent straight to the first input of its M_C subroutine, but if we were to also send the second duplicate of $< M >$ straight to the second input of M_C, it would not be well-formatted because M_C is expecting an n parameter. The second duplicate must therefore be modified slightly, so before proceeding, we first send the second copy of $< M >$ to a preprocessing subroutine R that appends the appropriate unique character "," as well as n so that this string is now of the form $< M, n >$, where n is a hard-coded constant that will be defined momentarily.

The inputs to M_C are now correctly formatted, so it proceeds to compute (itself using strictly fewer than n steps) whether or not consciousness is achieved by M's execution on s within n steps. If not, we

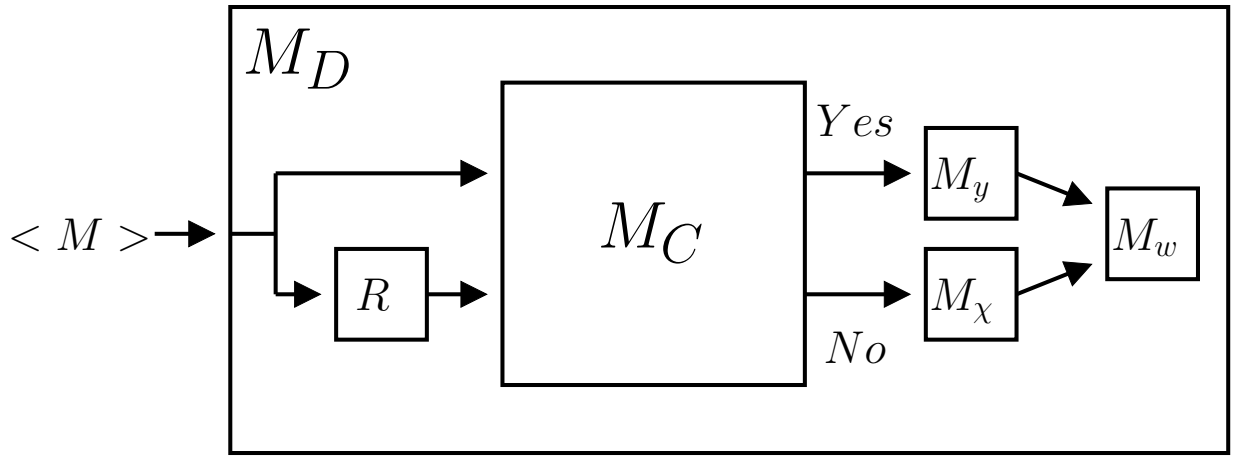

Figure 4.4: Schematic of M_D for Theorem 4.4

execute subroutine M_χ, which by Definition 4.1 is the machine which is able to achieve consciousness in the least number of steps χ. If so, then we execute subroutine M_y which simply wastes χ steps without achieving consciousness. Finally, regardless of whether M_χ or M_y was executed, we run M_w, which has a specific task: M_C is allowed at most $n - \chi - p$ steps to complete its computation, but if it finishes early, then M_w unconsciously wastes exactly the number of additional steps required to reach this limit.

We will derive a contradiction by inputting $< M_D >$ and $< M_D, n >$ to M_C which is the same as running M_D on $< M_D >$ for n steps and showing that these two characterizations of the same phenomenon give opposite results. We therefore need R to choose n such that both of these descriptions are executing for precisely the same number of steps. By construction, M_D will require precisely $c+\chi+p$ steps to execute where c is the number of steps used together by M_C and M_w, χ is the number of steps required by either M_χ or M_y, regardless of which one executed, and p is the number of steps required by the rest of the 'plumbing' of M_D to perform all of the remaining processing. From the statement of Theorem 4.4, $c = n - \epsilon = n - \chi - p$, so M_D will require precisely $n - \chi - p + \chi + p = n$ steps by construction.

Now let us determine the value of n output by subroutine R: By the statement of Theorem 4.4, M_C is allowed at most $n - \epsilon = n - \chi - p$ steps, so n it must be equal to at least $\chi + p$ to be non-negative. We can choose $n - \epsilon$ to be any arbitrary non-negative value, so for emphasis, let us set R's definition of n to be the deliberately large constant $n = \chi^{100} + p$. Note that this hard-coded value of n being input into M_C is an independent constant and that M_χ is the only component of M_D that is capable of consciousness because M_C cannot be be conscious by the assumption at the beginning of the proof, and By Definition 3.2, all of the remaining plumbing in M_D, including R, M_y, and M_w are mundane and do not give rise to consciousness. It is also worth emphasizing the distinction between n, which is the number of steps that $< M >$ executes on $< s >$, and $n - \epsilon$, which is the limit of the number of steps allowed to M_C.

Theorem 4.4. *If Assumption 3.1 holds true, then for all $n > \chi + p$, it is not possible to build an N-Step AMCD M_C described in Definition 4.2 that computes whether M is conscious when run on input s for n steps where M_C is not itself capable of consciousness and can itself can use at most $n - \epsilon$ steps. The constant $\epsilon = \chi + p$, where the constant χ is described in Definition 4.1, and the constant p is equal to the 'plumbing' steps of M_D as described in the previous paragraphs.*

Proof: Suppose that Assumption 3.1 holds true, and assume that it is possible to build an N-Step *AMCD* M_C as described in Definition 4.2 that computes its output within $n - \epsilon$ steps such that M_C is

not capable of consciousness, where $\epsilon = \chi + p$. We can therefore build M_D as described above.

Let us take an N-Step $AMCD$ 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 inputs $< M_D >$ and $< M_D, n >$ where $n = \chi^{100} + p$. M_C^* will tell us whether M_D running on $< M_D >$ for precisely n steps is conscious or not by respectively outputting 'Yes' or 'No', and M_C^* itself has at most n - χ - p steps to perform this calculation. Let us examine both cases by tracing the execution of M_D on $< M_D >$ according to Figure 4.4:

Case 1: Suppose M_C^* outputs 'Yes', which means that M_D running on $< M_D >$ for n steps is conscious. We input $< M_D >$ into M_D, where it is duplicated. The upper branch is sent to subroutine M_C, and the lower branch is sent to subroutine R, which converts the second duplicate of $< M_D >$ to $< M_D, \chi^{100} + p >$. Because M_C^* running on $< M_D >$ and $< M_D, \chi^{100} + p >$ output 'Yes', so does subroutine M_C. M_y and M_w are then executed. By construction, M_D running on $< M_D >$ takes precisely $c + \chi + p = n - \epsilon + \chi + p = n - \chi - p + \chi + p = n$ steps, where n is the same input given to M_C^*. However, M_χ didn't execute, and this was the only component capable of consciousness, so M_D was not conscious while running on $< M_D >$ for n steps, thus contradicting the output of M_C^*.

Case 2: Suppose M_C^* outputs 'No', which means that M_D running on $< M_D >$ for n steps is not conscious. We input $< M_D >$ into M_D, where it is duplicated. The upper branch is sent to subroutine M_C, and the lower branch is sent to subroutine R, which converts the second duplicate of $< M_D >$ to $< M_D, \chi^{100} + p >$. Because M_C^* running on $< M_D >$ and $< M_D, \chi^{100} + p >$ output 'No', so does subroutine M_C. M_χ and M_w are then executed. By construction, M_D running on $< M_D >$ takes precisely $c + \chi + p = n - \epsilon + \chi + p = n - \chi - p + \chi + p = n$ steps, where n is the same input given to M_C^*. However, M_χ executed, so M_D was conscious while running on $< M_D >$ for n steps, again contradicting the output of M_C^*.

In both cases M_C^* produced the wrong answer, so our assumption that it is possible to build an unconscious version of M_C that computes its output within $n - \epsilon = n - \chi - p$ steps is false. Therefore under Assumption 3.1, if it is possible to build an N-Step $AMCD$ M_C that computes its output within $n - \epsilon$ steps, then it must be conscious, as required. □

The careful reader will once again note that in the proof of Theorem 4.4, one can substitute virtually any non-trivial property P of a Turing Machine instead of consciousness and the proof will still hold - for example, it is not difficult to modify this result to show that the time-bounded N-Step Halting Problem is not computable. As was the case with Theorem 4.2, this generality is a feature rather than a flaw because it allows us to solve the problem of not having a sufficiently precise definition of consciousness and consequently this result holds for any reasonable definition of consciousness. We can generalize Theorem 4.4 to hold for all qualia with the following Corollary:

Corollary 4.3. *If Assumption 3.1 holds true, then for all qualia Q and all $n > \chi + p$, it is not possible to build Turing Machine M_Q that takes as input the encoding of another Turing Machine $< M >$ as well as the encoding $< s >$ of an input to $< M >$ and computes whether M experiences Q when run on input s for n steps where M_Q is not itself capable of experiencing Q and can itself can use at most $n - \epsilon$ steps. The constant $\epsilon = \chi + p$, and the constants χ and p are analogous to those in Theorem 4.4.*

4.4 Interpretation of Results

The three results above constitute both computability and incomputability results for different classes of $AMCD$s. By discovering boundaries separating computable and incomputable $AMCD$s and proving results straddling both sides of them, we have been able to help bring these borders into focus, and some patterns are starting to emerge:

The problem of building Machine Consciousness Detectors seems to parallel the Halting Problem, where general version is not computable, whereas the N-Step version is. Theorem 4.2 tells us that unconscious $AMCD$s are impossible, while Theorem 4.4 shows that unconscious, time-bounded, N-Step $AMCD$s are impossible. Meanwhile, according to Corollary 4.2, conscious, time-unbounded N-Step $AMCD$s are possible. The fact that we have two classes of unconscious $AMCD$s that won't work, and

one class of conscious $AMCD$s that does work suggests that the consciousness of the detector itself may be an important or even necessary element. In the field of consciousness studies where it is so difficult to prove concrete and absolute mathematical results, this constitutes 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 that isn't conscious, because those attempts are guaranteed to fail. One might say that we live in a Universe where when it comes to recognizing consciousness, it "takes one to know one" (if recognizing consciousness is possible at all).

Similarly, our results suggest that the time bounds granted to an N-Step $AMCD$ to do its work may be critical. Theorem 4.3 showed that if we give an N-Step $AMCD$ the freedom to fully simulate the machine it is being tasked to inspect for all n steps, then it is possible to build such an $AMCD$. However, in the proof of Theorem 4.4, we set $n = \chi^{100} + p$, which was purposefully chosen for emphasis to be an absurdly large number (and could have been arbitrarily chosen to be even larger), but instead of granting M_C all n steps, we restrict it to $n - \epsilon = n - \chi - p$ steps, which is only an asymptotically small fraction shy of n. By Theorem 4.3, if we allow an N-Step $AMCD$ all n steps, then the problem would have been solvable, so denying it even a small constant number of steps may be enough to thwart its ability to detect machine consciousness, suggesting an extreme threshold sensitivity in this area.

There is another interpretation if we view these results through the lens of simulation. By Assumption 3.1 and Corollary 4.2, simulated consciousness is itself conscious. In other words, denying an $AMCD$ the ability to be conscious may simply be denying it the ability to simulate $< M >$ on s. Theorems 4.2 and 4.4 both disallowed consciousness, and therefore also denied the ability to use simulation as a tool. (Theorem 4.4 additionally denied the possibility of simulation by restricting n.) They both showed that a subsequent failure to create an $AMCD$ was inevitable. Meanwhile, Theorem 4.3 allowed simulation, and succeeded, so it could be the case that when detecting machine consciousness, there are no computational shortcuts, and simulation may be a necessary tool, although this is more likely to be the case with the N-Step version of the problem.

More research is required to fully understand these relationships and boundaries. Several conjectures and related avenues of research are described below in Section 6.

5 Concluding Remarks

How will we ever know if an AI is conscious? This question lies at the intersection of computer science, philosophy, as well as ethics, and significant advances in AI during 2022 - 2023 have made it more important than ever. This paper does not presume to take a position whether machine consciousness is possible. We do not know if the basic assumptions listed above are true, but if they are, then we have been able to show that different formulations of the $AMCD$ problem are computable and not computable.

Our results attempt to shed light on the inherent limitations of identifying and mastering machine consciousness. We have discovered boundaries separating the computability and non-computability of building different forms of $AMCD$s and have identified candidates for their characteristics that may be critical. In the general case of an $AMCD$ as described in Definition 3.4, we have shown that it is necessary for this $AMCD$ to itself be capable of consciousness, if it is possible to build one at all. This result is general and holds in all cases. If machine consciousness is possible and not ubiquitous as the panpsychists believe, then no matter what, nobody even in the distant future, regardless of how advanced their technology is, will ever be able to build an infallible $AMCD$ that itself is not capable of consciousness.

In the case of N-Step $AMCD$s described in Definition 4.2, we have shown that the consciousness of the detector, the time that it is given to produce its output, as well as the ability to simulate are all boundary conditions that can be toggled in order to place the problem on one side of computability or the other. Unfortunately, our positive result showing that it is possible to build an N-Step $AMCD$ is not immediately actionable because the assumptions underpinning it would require us to solve the 'Hard Problem of Consciousness' and also make considerable advancements in our computer simulation

capabilities.

It is worth noting that the present results only apply to computational means of detecting consciousness. As described above in Section 1, science knows almost nothing about the nature of consciousness, including its underlying causal mechanisms. We don't know how to detect it, or even if it can be detected in an objective way - indeed, the present paper provides grounds for skepticism. This lack of knowledge on our part lies at the heart of the 'Hard Problem of Consciousness'. Does consciousness only arise as a consequence of the configuration of certain materials, as is the case with magnetism? Or is it substrate independent? It is conceivable that consciousness is a physical property associated with matter, like mass or magnetism, and it can there be detected by some physical means that we don't yet understand. Perhaps when present, consciousness creates a 'mental field' analogous to a magnetic or a gravitational field, or a form of radiation that we haven't yet discovered, in which case something akin to an (unconscious) Geiger counter for this type of radiation would be all that is needed to detect machine consciousness. The present results rule out certain computational possibilities for machine consciousness detection, but as unlikely as these other physical possibilities may seem, the present results do not rule them out.

It is gratifying to prove results that unify two of Turing's great interests, namely the areas of computability and machine consciousness. Why did Turing himself not combine the two areas back in the 1950s and instead chose to devise a more subjective version of the Turing Test? The proof of Theorem 4.2 above so closely parallels Turing's own proof of the Halting Problem that one is tempted to conclude that without the hindsight benefit of fundamental papers such as [Arb61, Put67], his attention simply wasn't focused in this direction, possibly for lack of a technical definition of consciousness, or possibly because the idea of conscious software would have been too exotic so many decades ago.

Computer science and AI have progressed considerably since then, but much more research is still needed in order to solve the biggest problems mentioned above and gain true insights into the nature of consciousness.

6 Related Open Problems, Conjectures, & Future Research

Rather than focusing on the major open problems in the area of consciousness studies, we will highlight some more tactical research areas that are closely related to the present results:

6.1 Generalization of The AMCD Problem

It is tempting to conjecture that the obvious generalization of Theorem 4.2 is true. In this result, we were only able to show that one cannot detect machine consciousness with an *unconscious* $AMCD$, but in all likelihood it isn't possible to create an $AMCD$, whether conscious or not, because the detection of almost any non-trivial property of a Turing Machine by another machine is typically not computable. As discussed above in Section 4.1, Rice's Theorem technically does not apply here, but that does not mean that it cannot guide our intuitions. Together with Theorem 4.2, this suggests that the following conjecture is likely true, and proving it would constitute a valuable extension of the present research:

Conjecture 6.1. *If Assumption 3.1 holds true, then it is not possible to create an Automated Machine Consciousness Detector M_C, whether it is capable of consciousness or not.*

6.2 Generalization of The N-Step AMCD Problem

More research into the computability of N-Step $AMCD$s is also needed. In Theorem 4.4, we simultaneously made the problem more difficult in two ways, by both disallowing M_C to be conscious, and also by restricting the number of steps it is allowed to perform below n. The Scientific Method dictates that we should change only one variable at a time, so it would be preferable to prove this result with only one change or the other. Since both of these changes remove the ability to use simulation as a tool, it is tempting to conjecture that both resulting problems would remain incomputable.

If we allow M_C to be conscious, then is it possible for it to do its work in under n steps? That is the purpose of the following conjecture:

Conjecture 6.2. *Under the premise that Assumption 3.1 holds true, it is not possible to build an N-Step Automated Machine Consciousness Detector M_C that computes its output within $n - \epsilon$ steps, where n is described in Definition 4.2, and ϵ is a constant.*

Conversely, is this problem solvable if we disallow M_C from being conscious, but do allow it the full n steps in which to do its work? Again we hypothesize that the answer is no:

Conjecture 6.3. *Under the premise that Assumption 3.1 holds true, it is not possible to build an N-Step Automated Machine Consciousness Detector M_C that is itself not capable of consciousness.*

6.3 Implications For The Physical Nature of Consciousness

Another research direction is suggested by our positive result in Theorem 4.3. In it, we connected the physical and theoretical worlds through computer simulation. Can this technique be used to 'boost' negative theoretical results into the real world and thereby allow us to draw conclusions about the physical nature of consciousness? More specifically, if we assume that the physical nature of consciousness is such that it can be detected by some type of physical sensor S, and if we can use a simulated form of S to build a consciousness detector in a simulation that contradicts a known incomputability result, then that contradiction in the simulated digital world would mean that the nature of consciousness in the physical world is such that it cannot be detected by any physical sensor. Any result along these lines ruling out such possibilities would constitute a *major* breakthrough in our understanding of consciousness and progress towards solving the Hard Problem.

Acknowledgments

TODO: THANK EVERYONE WHO PROVIDED COMMENTS

References

[Aar13] S. Aaronson. The Ghost In The Quantum Turing Machine. URL: https://www.scottaaronson.com/papers/giqtm3.pdf, 2013.

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

[BB22] L. Blum and M. Blum. A Theory of Consciousness From A Theoretical Computer Science Perspective: Insights From The Conscious Turing Machine. *Proceedings of The National Academy of Sciences of The United States of America (PNAS)*, Vol. 119, No. 21, 2022.

[Cha96] D. Chalmers. *The Conscious Mind: In Search Of A Fundamental Theory.* Oxford University Press, New York, 1996.

[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.

[Fag18] D. Faggella. Could Artificial Intelligence Become Conscious? 33 Researchers Contribute Their Opinion. URL: https://emerj.com/ai-market-research/conscious-artificial-intelligence/, 2018.

[Hai19] P. O. Haikonen. *Consciousness and Robot Sentience, Second Edition.* World Scientific Publishing Co., Singapore, 2019.

[Hai20] P. O. Haikonen. On Artificial Intelligence and Consciousness. *Journal of Artificial Intelligence and Consciousness*, Vol. 7, No. 1:73 – 82, 2020.

[Har19] A. Harris. *Conscious*. HarperCollins, New York, 2019.

[Low12] P. Low. Cambridge Declaration on Consciousness. In *Proceedings of the Francis Crick Memorial Conference*, pages 1 – 2. Cambridge University, 2012.

[McC79] J. McCarthy. Ascribing Mental Qualities to Machines. In M. Ringle, editor, *Philosophical Perspectives in Artificial Intelligence*, page 161. Humanities Press, Atlantic Highlands, N.J., 1979.

[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.

[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.

[Ric53] H. G. Rice. Classes of Recursively Enumerable Sets and Their Decision Problems. *Transactions of The American Mathematical Society*, Vol. 74, No. 2:358 – 366, 1953.

[Sip97] M. Sipser. *Introduction to The Theory of Computation, First Edition*. PWS Publishing Company, Boston, MA, 1997.

[Sip13] M. Sipser. *Introduction to The Theory of Computation, Third Edition*. Cengage Learning, Boston, MA, 2013.

[Tay99] C. C. W. Taylor. *The Atomists Leucippus and Democritus: Fragments, A Text and Translation with Commentary*. University of Toronto Press, Toronto, 1999.

[TK15] G. Tononi and C. Koch. Consciousness: Here, There, and Everywhere? *Philosophical Transactions of The Royal Society B*, Vol. 370, No. 1668:20140167, 2015.

[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.

[Tye21] Michael Tye. Qualia. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Fall 2021 edition, 2021.