Is the church turing thesis a red herring for cognitive science?

PETTERS, Dean and JUNG, A.

Published version

PETTERS, Dean and JUNG, A. (2021). Is the church turing thesis a red herring for cognitive science? In: AISB Convention 2021. Society for the Study of Artificial Intelligence and Simulation of Behaviour, 162-165.

Is the Church Turing Thesis a Red Herring For Cognitive Science?

Dean Petters [1] and **Achim Jung** [2]

Abstract. This paper considers whether computational formalisms beyond the Church Turing Thesis (CTT) could be helpful in understanding the mind. We argue that they may be, and that the way that the CTT has been invoked in Cognitive Science may therefore act as a Red Herring. That is, the way the CTT is invoked in Cognitive Science may mislead and perhaps contribute to premature abandonment of possibly fruitful research directions in Cognitive Science. We do not suggest some sort of "hypercomputation'. Whilst it is possible to use a rich interactive machine to implement a simple function this does not lead to new computable functions. In other words, the CTT is valid even if more sophisticated machinery is employed. It is the other direction that is the core of this paper: When considering more sophisticated computational tasks, then standard Turing machines (and their mode of operation) are not sufficient to explore the range of possibilities. The CTT is commonly interpreted as stating that the intuitive concept of computability is fully captured by Turing machines or any equivalent formalism (such as recursive functions, the lamba calculus, Post production rules, and many others). The CTT implies that if a function is (intuitively) computable, then it can be computed by a Turing machine. Conversely, if a Turing machine cannot compute a function, it is not computable by any mechanism whatsoever. We suggest an inadvertent error that has been made which is the claim that relatively simple computational formalisms like Turing Machines can do anything that more complex computional formalisms can do. To show this we present the landscape of computability within and beyond the bounds covered by the mathematical CTT. This shows that in regions of the computational landscape beyond the CTT there may be hierarchies of increasingly powerful computational formalisms. Erroneously interpreting CTT as enforcing a 'one size fits all' interpretation to computational formalisms leads to extreme reductionism that means contemporary computationalism is viewed as inadequate to explaining many phenomena related to thought and mind in living systems. Once this Red Herring interpretation for CTT is avoided this leaves the way open to exploring how richer kinds of computation that may possess many shades of expressivity can form part of Cognitive Science explanations.

1 Introduction

This paper takes the position that there are physically implementable programs which are outside the scope of the Church-Turing thesis (CTT). That is, we refute the existing idea that all computation has a boundary between what are computable functions and non-computable functions that is clear and distinct boundary for all formalisms. We show why this finding is important for Cognitive Science. The central argument of this paper is that invoking a mathematical theorem to make inferences about real-time physically instantiated systems should be done with careful consideration of both the scope of the theorem and the properties and complexity of the physical system. Turing set out to solve the "Entscheidungsproblem" (decision problem) and for this purpose proposed a mathematical formalism that faithfully emulates the process of a human being following finitely specified instructions. It was soon found that other formalisms have the same expressive power in this specific setting, i.e., mathematical problem solving, and this then led to the CTT. Situations in contemporary computing are now so rich, they can no longer be said to be covered by a paradigm where the inputs are known in advance, the system is left alone to do its computation and then provides the answer. Critically, for richer kinds of computation, the empirical evidence suggests that there are many shades of expressivity, which is why no-one has ever postulated an analogue of the CTT for them. This has implications for Cognitive Science and Artificial Intelligence. This is because it means that there may be computational formalisms which are strictly beyond the existing CTT but nevertheless recognisably symbolic/representational (GOFAI) in approach. Thus showing that cognitive scientists do not need to 'go all the way' to invoke non-representational or non-computational approaches (so called nouvelle AI such as enactivist [5], embodied [10], or dynamical systems approaches [11]) when going beyond classic computational formalisms. Instead, to explore how recognisably computational (representational/symbolic) systems can model phenomena of interest differently to formalisms that are within the scope of the CTT they only need to go 'slightly' beyond CTT and keep within the realm of computationalism. In discussing formalisms beyond CTT we not proposing a form of hypercomputation. The CTT is still valid when more sophisticated machinery is employed when that machinery is used to do carry out computational tasks that can be carried out by a Turing Machine (or computationally equivalent formalism). We are instead considering more sophisticated computational tasks than standard Turing machines (and their mode of operation) are sufficient to explore. This is of critical relevance to Cognitive Science - which studies humans performance on such sophisticated tasks to discover what computational formalisms humans possess. These formalisms may be beyond the CTT but still be symbolic computation.

2 The role of the CTT in Computationalism in Cognitive Science

The CTT is closely linked to the historical origins of computationalist (cognitivist) account of cognition. For example, in his historical review Clarke [2] notes:

[1] Sheffield Hallam University, UK, email: d.petters@shu.ac.uk
[2] University of Birmingham, UK, email: A.Jung@cs.bham.ac.uk

"The next big development was the formalization (Turing, 1936) of the notion of computation itself. Turing's work, which predates the development of the digital computer, introduced the foundational notion of (what has since come to be known as) the Turing machine. This is an imaginary device consisting of an infinite tape, a simple processor (a "finite state machine"), and a read/write head. The tape acts as a data store, using some fixed set of symbols. The read/write head can read a symbol off the tape, move itself one square backward or forward on the tape, and write onto the tape. The finite state machine (a kind of central processor) has enough memory to recall what symbol was just read and what state it (the finite state machine) was in. These two facts together determine the next action, which is carried out by the read/write head, and determine also the next state of the finite state machine. What Turing showed was that some such device, performing a sequence of simple computations governed by the symbols on the tape, could compute the answer to any sufficiently well-specified problem. "

"We thus confront a quite marvelous confluence of ideas. Turing's work clearly suggested the notion of a physical machine whose syntax following properties would enable it to solve any well-specified problem." ([2], p. 11-13)

In this historical analysis Clarke suggests that the concept of Turing Machines and the CTT, along with ideas that had previously been formulated on logics and formal systems, led to a radical new computationalist approach in Cognitive Science. Clarke cites Pylyshyn, who made these same points in the 1970s:

"The work of Turing, in a sense, marked the beginnings of cognitive activity from an abstract point of view, divorced in principle from both biological and phenomenological foundations. It provides a reference point for the scientific ideal of a mechanistic process which could be understood without raising the spectre of vital forces or elusive homunculi but which at the same time was sufficiently rich to cover every conceivable formal notion of mechanism (that the Turing formulation does cover all such notions is, of course, not provable but is has stood all attempts to find exceptions. The belief that it does cover all possible cases of mechanism has become known as the Church-Turing thesis). It would be difficult to overestimate the importance of this development for psychology. It represents the emergence of a new level of analysis, which is independent of physics, yet it mechanistic in spirit. It makes possible a science of structure and function divorced from material substance, while at the same time it avoids the retreat to behavioralistic periperheralismm. It speaks the language of mental structures and of internal processes, thus lending itself to answering questinos traditionally posed by psychologists"

"While Turing and other mathematicians, logicians, and philosophers laid the foundations for the abstract study of cognition in the 30s and 40s it was only in the last twenty or so years thatthis idea begain to be articuated in a much more specific and detailed form: A form which lends itself more directly to attacking certain basic questions of cognitive psychology. The newer direction has grown with the continuing development of our understanding of the nature of computational process and of the digital computer as a general, symbol-processing system. It has led to the formation of a new intellectual discipline known as artificial intelligence, which attemnpts to understand the nature of intelligence by designing computational systems which exhibit it" ([8], 24-25)

What these quotes show is how the CTT led to promotion of multiple realisability and the stronger notion of medium independence as supporting foundations for cognitive science. However, gaining the notion of multiple realisability through invocation of the CTT brought with it the possibility of a limiting misconception - a Red Herring - as this misconception that all computational formalisms are equivalent has led to a mistaken view of computational approaches to the mind leading to extreme reductionism. This extreme reductionism follows from the misconception that very simple computational formalisms are computationally equivalent to more complex formalisms because they can produce the same set of functions. Turing's original machine is a very simple abstract concept. There is a control unit in a particular state, and finitely many alternative states. There is also an infinite tape, which acts as a memory and on which can be marked '0','1' or 'nothing'. There is also a read-write head which takes decisions and can change a '0' to a '1', change a '1' to a '0', or erase a '0' or '1'. There are even simpler computational formalisms like the *two-counter machine*. This can increment and decrement with branching. The extreme reductionism becomes apparent when we ask: Can this or a Turing Machine be programmed to be conscious? The line of reasoning that acts as a Red Herring is: If consciousness arises from computation, and the CTT is correct in stating that all computational frameworks are equivalent, then if any computational system can exhibit self-awareness these kind of simple machines will exhibit self-awareness. The widespread view (that we agree with) that Turing Machines or two-counter machines cannot be conscious simply by running the right program has led to the conclusion that psychological phenomena such as consciousness, agency or self-awareness are not computational in origin. We suggest that a different route out of this impasse is to accept that applying CTT to all forms of computation is a Red Herring. That it, it is an unhelpful misconception that misdirects research. Researchers looking for computational explanations for complex psychological phenomena that are not simply function computations should look beyond the CTT to more sophisticated computational formalisms.

Not all researchers have been misdirected by a misconception that the CTT applies to all forms of computation. Goldin and Wegner [3], examine this misconception and suggest that the operation of "batch processing" in the first generation of computing machines was so similar to Turing's mathematical concept of a (human) "computer" (i.e., his "Turing machines") that Turing machines were incorrectly adopted as a sole formal abstraction of computing practice. Goldin and Wegner point out the role of interactivity in processes that is so central to modern computing system is simply not covered by the CTT. Some researchers have been very aware that Turing machines are not appropriate for modelling interactive processes and have proposed alternative mathematical abstractions [4, 6]. A key issue is that when we consider computation from fixed input to single output (the "function view" of computation), then the equivalence of computational mechanisms is almost unavoidable. In contrast, mathematical models for interactive behaviour (the "process view" of computation) can be *quite different* in expressivity. A canonical, maximally expressive formalism for processes simply does not exist. We point the interested reader to Abramsky's [1] where this fact is highlighted and explored.

3 The landscape of computability in diagrams

The landscape of computability includes regions within and beyond the bounds of the classic Church-Turing thesis. Figure 1 shows that for batch style computation (all of the formalisms on both the top and bottom of the left half of the diagram) there is a 'one size fits' organisation of the landscape of computability due to the CTT. That is, functions are either computable or non-computable whatever formalim is used. Regarding the right-hand side of the diagram: the question mark signifies that we don't know what the situation is. Before we can distinguish computable from non-computable entities on the right hand side of the diagram, we first need to decide what "entity" is being computed by a distributed or probabilistic or other kind of program. Once we have a clear idea for that (unlikely in the case of distributed computing), we can then try to see whether we get an analogue of the CTT (with all reasonable formalisms being computationally equivalent), or whether the situation is more like that of the total functions in the bottom left square of the left hand side of the diagram. That is, for total functions that are computable, different formalisms cover different parts of the computable realm and none covers all of it. Therefore, the right hand side of the landscape of computability (for contemporary and future computation) might have an infinite tower of increasingly powerful computational formalisms. Or some other kind of hierarchy. Such as a finite tower of increasingly powerful formalisms. This is important for Cognitive Science because it means there may be computational formalisms for contemporary and future computational approaches that do not have a 'one size fits all' organisation. Therefore perhaps changing the current extreme reductionism which is currently justified by invocation of the CTT. The rationale for this extreme reductionism is that all formalisms within the scope of the CTT, however complex, can produce the same set of computations as very simple formalisms like Turing Machines).

4 From the Church Turing Thesis to the Chinese Room Argument

Figure 2 situates particular kinds of programs in the landscape of computability. In particular, situating the kind of batch program that Searle describes in his Chinese Room Argument (CRA) [9] and a class of adapted CRA program sketched by Petters and Jung [7] - with interruptions and interactivity, real-time processing, never-ending computation and parallel distributed control [1, 3, 4, 6]. This adapted CRA program will not lead to new computable functions, i.e., some sort of "hypercomputation". The CTT is still valid when more sophisticated machinery is employed to compute functions that could be computed by programs within simpler formalisms. Our claim is that when considering more sophisticated computational tasks, standard Turing machines (and their mode of operation) are not sufficient to explore the range of possibilities that can be produced with this kind of formalism.

5 Conclusion

This paper argues that due to the extreme reductionism enforced by the Church Turing Thesis contemporary computationalism is inadequate to explaining many phenomena related to thought and mind in living systems. This paper is not proposing a kind of hypercomputation. Whilst it is possible to use a rich interactive machine to implement a simple function this does not lead to new computable functions'. In other words, the CTT is valid even if more sophisticated machinery is employed. It is the other direction that is the core of this paper: When considering more sophisticated computational tasks, then standard Turing machines (and their mode of operation) are not sufficient to explore the range of possibilities. This paper suggests computational formalisms beyond the computational formalisms covered by the mathematical Church Turing are likely to be particularly valuable for explaining cognitive processes in living organisms that are not simply function computations.

REFERENCES

[1] S. Abramsky, 'Intensionality, definability and computation', in *Johan van Benthem on Logic and Information Dynamics. Outstanding Contributions to Logic, vol 5.*, eds., A. Baltag and S. Smets, 121–142, Springer, (2014).

[2] A. Clark, *Mindware*, Oxford University Press,, Oxford, 2014. 2nd edition.

[3] D.Q. Goldin and P. Wegner, 'The Church-Turing thesis: Breaking the myth', in *New Computational Paradigms*, eds., S. Barry Cooper, Benedikt Löwe, and Leen Torenvliet, pp. 152–168. Springer Berlin Heidelberg, (2005).

[4] C.A.R. Hoare, *Communicating Sequential Processes*, Prentice Hall International, 1985.

[5] D. Hutto and M. Myin, *Radicalizing Enactivism: Basic Minds without Content*, MIT Press, Cambridge, MA, 2013.

[6] R. Milner, *A Calculus for Communicating Systems*, volume 92 of *Lecture Notes in Computer Science*, Springer Verlag, 1980.

[7] D. Petters and A. Jung, 'From the Chinese Room Argument to the Church-Turing Thesis', in *Proceedings of the International Symposium on 'Philosophy after AI: Mind, Language and Action*, eds., G. Gallo and C. Stancati, 25–29, AISB Press, University of Liverpool, (2018).

[8] Z. Pylyshyn, 'Complexity and the Study of Human and Machine Intelligence', in *Mind Design*, ed., J. Haugeland, 23–56, Bradford Books/MIT Press, Cambridge, MA, (1979).

[9] J.R. Searle, 'Minds, brains, and programs', *The Behavioral and Brain Sciences*, **3**(3), (1980). (With commentaries and reply by Searle).

[10] L. Shapiro, *Embodied Cognition*, Routledge, London, UK, 2010.

[11] T. van Gelder, 'Dynamics and cognition', in *Mind Design II*, ed., J. Haugeland, MIT Press, Cambridge, MA, (1997).

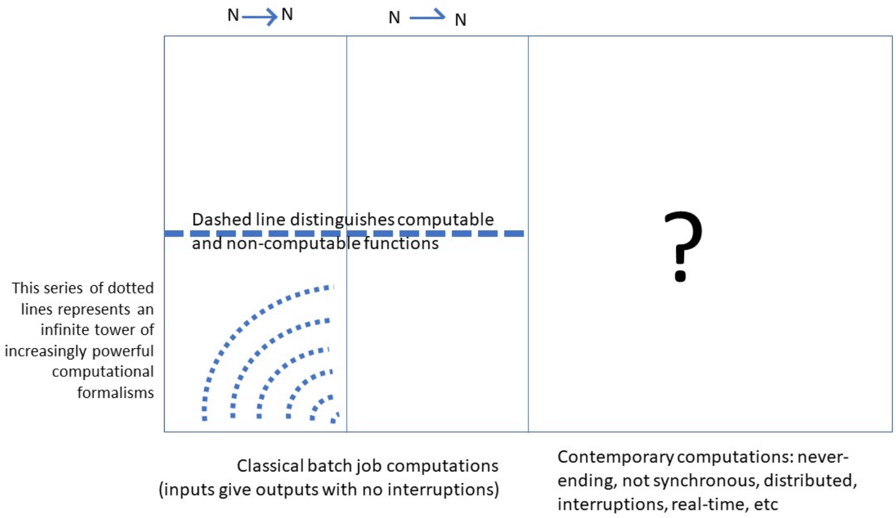

Figure 1. The landscape of computability. There are two vertical lines in the diagram. The left line distinguishes between total and partial functions. In the region of the landscape where total functions are computable it can be shown that for whatever formalism is considered, diagonalization can always be used to create a new function beyond the set computable by that formalism. So there is an infinite number of possible formalisms forming a tower of increasing computational power - represented by a series of curved dotted lines. The right hand split distinguishes classical batch computation or not. The right hand side of the right hand vertical line is therefore contemporary approaches like never-ending, not-synchronised, distributed, real-time, and probabilistic computation. As well as computations with other attributes we take for granted in 2020. Plus as yet undiscovered models possessing attributes that are beyond what can be possessed by computations in Turings and other formalisms for computation (like Churchs, Posts, Gödel's and many others). The dashed horizontal line through the left half of the figure distinguishes computable and non-computable functions. According to CTT, every reasonable formalism gives all lower left outputs below the dashed line. There is (as yet) no comparable thesis to the Church-Turing thesis for the right of the figure (contemporary computation). (Note: the areas are not to scale, below the horizontal split (computable) would actually be a tiny sliver compared to non-computable.)

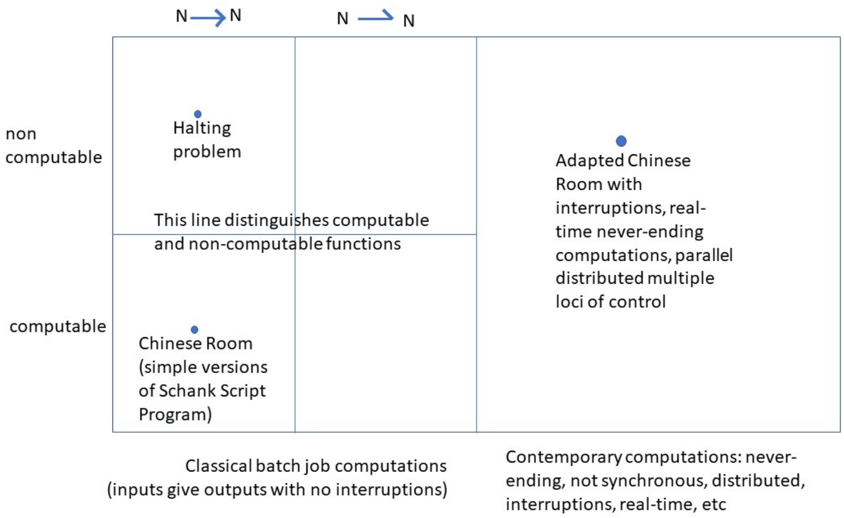

Figure 2. Situating particular kinds of programs in the landscape of computability. Turing showed that the halting problem was non-computable. Petters and Jung [7] show that Searle's original CRA argument through experiment described a batch job program that was computable. Petters and Jung [7] also briefly sketch an adapted CRA program with interruptions and interactivity, real-time processing, never-ending computation and parallel distributed control which is outside the scope of the traditional CTT and so on the right hand side of the landscape of computability. So this adapted CRA program may or may not be computable, and the formalism used to implement it may be more computationally powerful than the formalism used to implement Searle's original chinese room program. Therefore, conclusions from the original CRA may not apply to all implementable programs.