From the Chinese Room Argument to the Church-Turing Thesis

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

Abstract. Searle's Chinese Room thought experiment incorporates a number of assumptions about the role and nature of programs within the computational theory of mind. Two assumptions are analysed in this paper. One is concerned with how interactive we should expect programs to be for a complex cognitive system to be interpreted as having understanding about its environment and its own inner processes. The second is about how self-reflective programs might analyse their own processes. In particular, how self-reflection, and a high level of interactivity with the environment and other intelligent agents in the environment, may give rise to understanding in artificial cognitive systems. A further contribution that this paper makes is to demonstrate that the Church-Turing Thesis does not apply to interactive systems, and to self-reflective systems that incorporate interactivity. This is an important finding because it means that claims about interactive and self-reflective systems need to be considered on a case by case basis rather than using lessons from relatively simple non-interactive and non-reflective computational models to generalise to all computational processes.

1 Introduction

This paper will show that Searle's Chinese Room Argument (CRA) scenario [14] can be extended and given more detail so that new variations of this scenario have a fundamentally different relationship with the Church-Turing Thesis (CTT). Searle's CRA is a gedanken experiment aimed at demonstrating that computer programs cannot really understand the meaning of what they process, even if their observable behaviour seems to demonstrate understanding. 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.

This paper presents a family of variations to the CRA which involve changing the CRA to require significantly more interaction with the outside world: in frequency of interruptions; interleaving of interruptions; and in the nature of the information provided by interruptions. A second family of variations to the CRA includes the same pattern of interruptions and close coupled interaction with the external environment but also includes ways in which higher-level routines within the CRA program can analyse the basic program for 'meaningful' patterns in its own internal processing. These versions of the CRA are outside the scope of the CTT because the CTT is concerned with situations where programs act as mathematical functions with inputs fully provided at the start of the computation and with no possibility of new inputs being included during the run of the program. This matters because the CTT is commonly invoked to generalise the lessons from the CRA to *all forms of computation whatsoever* while it is only legitimate to draw conclusions about programs and computational mechanisms which follow the basic input-output paradigm. If programs presented in new variants of the CRA scenario fall outside the scope of the CTT (but are still recognisable and implementable programs in the sense of being precisely specifiable algorithms) then the lessons from these variants will not necessarily generalise to all possible programs. Therefore, any generalisations would need to be validated on a case-by-case basis for prospective program formalisms. The paper concludes with the observation that the new variant CRA scenarios sketched in this paper are not only more similar to typical human cognition than the very simplified portrayal of processing in the original CRA, but the complexity they present is fast being achieved and overtaken by contemporary computing systems.

2 Overview of the CRA — and how lessons drawn from it are generalised

Published in 1980 in the paper *"Minds, Brains, and Programs"*, [14], Searle made an argument based on a 'Chinese Room'. It is a thought experiment that is intended to show that running programs cannot have understanding and awareness of what they are doing. Searle introduced his first CRA scenario by discussing an earlier simulation produced by Roger Schank and co-workers, [13]. Searle explained that he was using that work as inspiration for this CRA scenario because of his own familiarity with this program. However, he also claimed that his argument does not rely on the details of Schank's programs, and in fact applies to any Turing machine simulation that is modelling mental processes. It is this claim of generalisation to all programs (because Schank's program can be run on a Turing machine) that is the critical focus of the present paper.

Schank's program simulates the ability to understand stories. The program accesses information about particular contexts and the program can then answer questions about a story set in that context. This is accomplished by analysing what is stated in the story and what can be expected in the context in which this particular story is set. Schank's program accomplishes this by possessing a representation, which he terms a 'script' that includes contextual information of the sort that humans possess. Searle highlights the fact that in this process it is only the form of the representations of the story and script that are necessary and sufficient to produce the output. The content of the representations of the story and script takes no part in the algo-

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

rithmic process and is not required to transform input to output. The key lesson that Searle draws from the CRA is that the formal symbol manipulations carried out within the Chinese room do not give rise to meaning or understanding, operations in the Chinese room are all *"syntax but not semantics"* ([14], p. 422).

3 Different varieties of Searle's Chinese Room scenario have fundamentally different relationships with the CTT

3.1 Searle's original scenario

> *'Suppose that I'm locked in a room and given a large batch of Chinese writing. Suppose furthermore (as is indeed the case) that I know no Chinese, either written or spoken, and that I'm not even confident that I could recognize Chinese writing as Chinese writing distinct from, say, Japanese writing or meaningless squiggles. To me, Chinese writing is just so many meaningless squiggles.*
>
> *Now suppose further that after this first batch of Chinese writing I am given a second batch of Chinese script together with a set of rules for correlating the second batch with the first batch. The rules are in English, and I understand these rules as well as any other native speaker of English. They enable me to correlate one set of formal symbols with another set of formal symbols, and all that 'formal' means here is that I can identify the symbols entirely by their shapes. Now suppose also that I am given a third batch of Chinese symbols together with some instructions, again in English, that enable me to correlate elements of this third batch with the first two batches, and these rules instruct me how to give back certain Chinese symbols with certain sorts of shapes in response to certain sorts of shapes given me in the third batch. Unknown to me, the people who are giving me all of these symbols call the first batch "a script," they call the second batch a "story". and they call the third batch "questions". Furthermore, they call the symbols I give them back in response to the third batch "answers to the questions" and the set of rules in English that they gave me, they call "the program." '* ([14], p. 418)

Searle's lesson is that an observer external to the room would see meaningful behaviour but within the room there is only meaningless symbol processing — so demonstrating that understanding cannot arise from just the operation of formal syntactic processes.

We can see that this very abstract description of a running program is not only based on a single run of the Schank program, but also matches the classic modus operandi of the very early generations of electronic computers. These carried out 'batch jobs' where the input and program were both completely specified at the start. The computing machine would process the input according to the program, and the output would appear as a paper printout. Contemporary computing no longer works like this, with many possible interruptions to ongoing processing. The next scenario attempts to sketch out different ways in which interruptions and new input data can appear during the running of a program.

3.2 Searle managing multiple tasks by effectively processing real-time updates from the environment

This quote from Monsell highlights the delicate balancing act in natural systems between forcing through ongoing processing on a primary task and dealing appropriately with potential interruptions:

> *"Hence the cognitive task we perform at each moment, and the efficacy with which we perform it, results from a complex interplay of deliberate intentions that are governed by goals ('endogenous' control) and the availability, frequency and recency of the alternative tasks afforded by the stimulus and its context ('exogenous' influences). Effective cognition requires a delicate, 'just-enough' calibration of endogenous control that is sufficient to protect an ongoing task from disruption (e.g. not looking up at every movement in the visual field), but does not compromise the flexibility that allows the rapid execution of other tasks when appropriate (e.g. when the moving object is a sabre-toothed tiger)."* ([10], p. 134).

Following Monsell, an interactive CRA scenario could capture the closely coupled nature of interactions between agent and environment and might involve running many sub-programs in parallel with an overarching program acting as a kind of operating system. Not only does Searle's 1980 scenario completely ignore the nature of algorithms that require this level of constant checking the current state of the environment, it also ignores the nature of 'forever' processes such as operating systems carrying out processes such as resource management and process control, but then returning to the same ground state and never providing a final output. Yet it seems something like this kind of 'operating system' algorithm must be implemented in humans and other animals. In addition, the ever increasing complexity of artificial control systems like intelligent mobile robots and self-driving cars can increasingly be seen to incorporate these kinds of complex interactions, driven by environment interruptions alongside the requirements of multiple primary tasks. So adding further complexity, interactive variants of the CRA scenario might also include parallel computation, probabilistic computation, and real-time computation, all of which are manifestly outside the scope of the CTT.

This leads to a key claim of this paper — an interactive variant of the Chinese Room Argument scenario similar to Monsell's description, getting input during the running of the program, but also including multi-processing, real-time computing, (truly) probabilistic computation, programs that never terminate, distributed computation, intentional computation, and higher order computation, is outside the scope of the CTT. This is because the CTT is concerned only with the equivalence of systems that operate from input provided as a string to output as a string; it does not cover programs that deal with a series of inputs, appearing over time and at unexpected moments, or that are capable of making changes to the operation of the program while it is running. As laid out clearly and carefully in his 1936 paper, Turing's concern was with the steps a mathematician (a human "computer" in Turing's terminology) goes through when following a precisely and finitely specified procedure. His analysis is compelling and we see no reason not to accept the CTT, though we emphasise its constrained setting. The (partial) functions from strings to strings that can be computed by Turing's "machines" are the same ones that can be computed by any and all formalisms that have to date been put forward as alternatives. Put another way, the evidence for the CTT is very strong indeed, but this does not give licence to applying it — by analogy, as it were — to other forms of computation. These "other forms of computation" are not the fruits of idle speculation but very much day-to-day reality for software engineers and computer users alike: computing machines no longer expect a question on a tape (or punch cards), go away and compute, and return an answer as a single file (or more punch cards or some output paper). Contemporary programs interact with their environment in multiple ways, and employ facilities (such as hardware-based random number generators) that can not be shoehorned into the paradigm for which the CTT was formulated.

4 The CTT in theoretical computer science and its implications for the philosophy of mind

4.1 Functions versus processes

The misconception that the CTT applies to all forms of computation is very wide-spread also within the computer science community, and even among theoretical computer scientists. Goldin and Wegner, [6], examine the likely origins of this belief, which they term the 'Strong Church-Turing Thesis', and explore the reasons why it holds such sway. They suggest that the way the first generation of computing machines were designed and used (i.e., the "batch processing" discussed above) was so strongly correlated with Turing's mathematical concept of a (human) "computer" (i.e., his "Turing machines") that standard undergraduate textbooks adopted Turing machines as a suitable formal abstraction of computing practice. Like us, Goldin and Wegner point out the role of interactivity that is so central to modern computing systems, and that is simply not covered by the CTT.

Some researchers have been very aware that Turing machines are not appropriate for modelling interactive behaviour and have proposed alternative mathematical abstractions. We mention especially the work of Milner, [9], and Hoare, [8], on computational "processes". It is astonishing (but not the focus of the present paper) that although their work has been incorporated into undergraduate syllabuses for decades, courses on computability theory still promulgate the view that Turing machines are all there is to computation.

Beyond the analysis for this state of affairs given in [6] we believe that it is useful for our argument to point out one crucial difference between the setting of the CTT and the more encompassing computational models of Milner and Hoare: When we consider computation from fixed input to single output (the "function view" of computation), then the equivalence of computational mechanisms is almost unavoidable. To give just one example, it is not the case that the λ-calculus was designed with computability in mind; rather, its purpose originally was to give a new foundation for mathematics, replacing set theory (see [2] for a historical introduction). As far as functions from natural numbers to natural numbers are concerned, the equivalence with Turing machine computability was noted *afterwards*. 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.

One final comment on the difference between the functional and the process point of view: It is, of course, possible to use a rich interactive machine to implement a simple function; after all, that is what we do with our modern computers all the time. It is our belief that this does not lead to new computable functions, i.e., some sort of "hypercomputation". 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.

4.2 Computation in an extended sense

So far, we have focused on interactivity as a (ubiquitous) feature of modern computational systems which is not present in the Turing machine model. There are others which are also interesting for our argument, especially in an interactive setting. We begin with the question whether the computational process has internal memory or not. If it does, then it can react differently to identical stimuli from the environment as time passes, and indeed it can exhibit "learning behaviour". A study of this facility from the point of view of computability theory is presented in [5], for example. What is important for our argument is the fact that an interactive process that has some finite internal memory is strictly more powerful than a process that does not, and a process that has unlimited internal memory is strictly more powerful than one with finite memory. Thus we have a fairly straightforward computational situation were the CTT is false, or to be more precise, where there is no analogue of the CTT.

If we translate these findings to Searle's Chinese Room, then we are in a situation where he may be in interaction with his environment, constantly receiving and issuing statements expressed in Chinese characters. Having the ability to make personal notes (in English but perhaps with Chinese characters interspersed) would greatly enrich his experience and might even lead him to understand the meaning of these interactions. This would be true even if the form of his note-taking were already prescribed in his original "script".

A similar argument can be made for processes that have access to a real-time clock, or to a source of true random numbers.

4.3 Implications for the philosophy of mind

Both Goldin and Wegner, and Abramsky, highlight an issue in theoretical science which has not yet been received in the philosophy of mind literature. They show that for computer scientists the CTT should be treated as a thesis that certain models of computation are equivalent for tasks that require the transformation of given fixed finite input to some output, and not that TMs can implement every possible kind of information processing machine. Goldin and Wegner do argue that researchers in Artificial Intelligence are somewhat ahead of researchers in theoretical computer science in promoting interaction rather than computation of functions as beneficial in expressing the behaviour of information processing systems. For example, they cite Rodney Brooks' 1991 statement of interaction as a prerequisite for intelligent system behaviour [3], and Russel and Norvig recognising that intelligent behaviour is better modelled by interactive agents than functions with prestated inputs and outputs that only occur at the termination of the computation [12]. However, this mistaken view, that the CTT states that TMs are capable of such broad information processing capabilities, seems to be what justifies the generalisation that lessons from Searle's specific CRA scenario applies to all possible programs. For example, in *The Critique of Cognitive Reason'* Searle invokes the CTT to state that for any algorithm there is a TM which can implement that algorithm — which is a correct interpretation of the CTT (assuming the common interpretation of "algorithm"). However, he then goes on to suggest that the next step from this line of reasoning is that the brain is a Universal TM. Whilst he concedes that in addition to algorithmic processes (within the scope of the CTT) there may be unconscious processes outside the scope of the CTT, he does not consider that processes which link up and transition between individual computations are of this unconscious type. In fact, he does not consider dynamic and contingent transitions between individual function-based computations at all ([15], p. 837).

4.4 Searle carrying out self-reflection of his own program

New interactive variants of the CRA may be outside the CTT, but they do not necessarily demonstrate more understanding in the inner workings of the Chinese room. Inserted information may be just as impossible for 'Searle in the room' to understand as the Chinese symbols in the original CRA scenario. However, we can not only vary frequency, interleaving, and parallelisation due to interruptions, but also form new CRA scenarios which involve kinds of information that are intended to interact with the running program to change the English rules that Searle carries out. The CTT does not cover processes where in principle any information (from a simple boolean to an analysis of the running of the existing program to a whole new program) can be added as input during the running of the program. In the book *'Kinds of Minds'*, Dennett [4] portrayed a number of different abstract agents (creatures) according to how they processed information. He presented Darwinian creatures as not capable of learning but acting upon evolved reflexes; Skinnerian creatures as learning from association; and Popperian creatures that can pre-select strategies after evaluating their likely success in internal working models. In addition to these creatures, Dennett also described Gregorian creatures, *"whose inner environments are informed by the designed portions of the outer environment."* ([4], p. 99). Thus Gregorian creatures can import 'mind tools' wholesale from the environment ([4], p. 100). What is relevant to the CRA and Searle's conclusion is whether the inputs to the Chinese room can not only add to the store of Chinese symbols but also add to or substitute for the English instructions that 'Searle in the room' actually follows. Programs which can be interrupted to receive new information that may alter in a fundamental way their processing, even conceivably by changing the running program itself, are clearly outside the scope of the CTT. This is because if a new program can be given as input on an interruption, this is no longer the program which started processing.

It is possible to have algorithms in the Chinese Room that engage in self-reflection and self-analysis. When self-reflection and self-analysis occur it can create a kind of internal 'meaning' about the system which may then be linked to external meaning in the form of patterns in Chinese symbols. Any 'Searle in the room' can only carry out the English instructions which are directly given to him. 'Searle in the room' can never do anything which is not part of a task set out in English instructions. But a 'Searle in the room' can follow the programmed instructions for the specific 'narrow' task at hand of processing stories, and his overall task can also involve a whole set of further instructions which may be triggered at any time, and often are triggered by well considered interruptions from the outside, and which involve questioning what the nature of the connections between internal rules and data mean. He can be asking, in addition to what patterns in input and output data exist, what patterns exist in the use of his English rules. When do rules co-occur? What rules predict other rules? Do some rules being triggered predict the task is nearly over? Are some patterns more surprising than others? Are there clusters or categories of rules that perform similar tasks? The 'Searle in the room' accomplishing this broader and self-reflective task is not carrying out a non-algorithmic process. Rather, he is still following English rules that compare rules and processes looking for identifiable patterns. But these patterns do not then trigger the outputting of meaningless symbols. Rather, Searle is learning about meaning in processing patterns apart and aside from the meaning of the symbol tokens being processed. Meaning is emerging from the internal processing of rules set apart from the meaning of the Chinese symbols.

These 'Searle in the room' self-reflection scenarios highlight distinctions between: (i) algorithms that carry out specific narrowly defined tasks, and just carry out those tasks versus algorithms that carry out tasks and simultaneously search for meaning in the properties and implications in patterns in running processes and events; and (ii) 'representational' meaning by understanding the content of symbol tokens versus 'dynamic processing' meaning by understanding the properties and implications in patterns in running processes and events.

Further variants of self-reflection scenarios include: 'Searle in the room' looking for meaning but making no connection to the external meaning of the Chinese symbols, so all meaning emerges from the bottom up; and, 'Searle in the room' looking for meaning internally and connecting this to appropriate external symbols — thus giving external symbols more than derived intentionality. He might be helped in this by people outside the room interacting with him in a way designed to foster the emergence of meaning.

5 Importance of the CTT for psychology and cognitive science

In his review of the literature around the CRA, Preston makes clear why CTT matters to psychology and cognitive science:

> *"Even more important than the nature of the thesis, perhaps, is the matter of its implications. It's no exaggeration to say that the Church-Turing thesis has constituted the fundamental inspiration behind AI, the reason for thinking that electronic digital computers must be capable of (at least) human-level intelligence. Cognitive scientists have generally taken the Church Turing thesis to mean that any function that can be computed can be computed by a Turing machine. This would mean that, as long as we ignore or abstract away from resource limitations, anything the human brain can do (any function it can compute) could also be done (computed) by an electronic digital computer. Cognitive processes, no matter how intelligent must be decomposed into routines whose primitive steps can all be executed by a machine"* ([11], p. 6).

Since it was first formulated in the 1940s no-one has really questioned the CTT, and nor do we. It is one of the jewels of theoretical computer science [7]. However, the CTT is concerned only with functions from from strings to strings — input needs to be given as a fixed finite string and output (if it is produced) will be a finite string. The CTT underlies the strength of the CRA because it allows Searle to say: 'the limitations of the program in the CRA applies to all programs because of the CTT'. Accepting this generalisation strategy, as Searle does, means there can be no syntactic formal processes in a program, of even fiendish complexity or strangeness, that will ever give rise when running to any kind of semantics. If, on the other hand, Searle cannot invoke the CTT for the CRA then whatever lessons he draws from the CRA only apply to the specific scenario he presents.

6 Conclusion

This paper takes the position that there are implementable programs which are outside the scope of the CTT. 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, some of which have been described in this paper, 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 paper therefore agrees with Searle insofar as when programs confirm to the requirements for CTT equivalence, there can be no meaning in the internal symbol processing. But for programs outside the scope of CTT, meaning can appear in several ways, (i), by interaction with the environment, and (ii), by self-reflection within the program. This paper challenges the idea that syntax (in the case of a running program) cannot give rise to semantics. Therefore this paper takes a radical approach which attempts to overturn 80 years of misguided extrapolation that the CTT applies to all programs that are of interest to computer science, cognitive science, and philosophy.

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] H.P. Barendregt, *The Lambda Calculus: Its Syntax and Semantics*, North-Holland, revised edn., 1984.

[3] R. Brooks, 'Intelligence without reason', in *Proceedings of the 12th International Joint Conference on Artificial Intelligence - Volume 1*, 569–595, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, (1991).

[4] D.C. Dennett, *Kinds of minds: towards an understanding of consciousness*, Weidenfeld and Nicholson, London, 1996.

[5] D.Q. Goldin, S.A. Smolka, P.C. Attie, and E.L. Sonderegger, 'Turing machines, transition systems, and interaction', *Information and Computation*, **194**, 101–128, (2004). Special Issue Commemorating the 50th Birthday Anniversary of Paris C. Kanellakis.

[6] 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).

[7] D. Harel and Y. Feldman, *Algorithmics: The Spirit of Computing*, Springer, 3rd edn., 2012.

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

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

[10] S. Monsell, 'Task switching', *Trends in Cognitive Science*, **7**, 134–140, (2003).

[11] J. Preston, 'Introduction', in *Views into the Chinese Room: New Essays on Searle and Artificial Intelligence*, ed., J. Preston, 1–50, Oxford University Press, Oxford, (2003).

[12] S. Russell and P. Norvig, *Artificial Intelligence, A Modern Approach, 2nd Edition*, Prentice Hall, 2003.

[13] R.C. Schank and R. Abelson, *Scripts, plans and understanding*, Lawrence Erlbaum Associates, Hove, UK, 1977.

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

[15] J.R. Searle, 'The critique of cognitive reason', in Readings in Philosophy and Cognitive Science*, eds. A. Goldman*, 833–847, MIT Press, Cambridge, (1993).