CHURCH'S THESIS
LOGIC, MIND AND NATURE

Edited by
ADAM OLSZEWSKI
BARTOSZ BROŻEK
PIOTR URBAŃCZYK

Copernicus Center PRESS

Editing:
Aeddan Shaw

Cover design:
Mariusz Banachowicz

Layout:
Mirosław Krzyszkowski

Typesetting:
Piotr Urbańczyk

ISBN 978-83-7886-009-9

Kraków 2014

Copernicus Center PRESS

Publisher: Copernicus Center Press Sp. z o.o.
pl. Szczepański 8, 31-011 Kraków
tel/fax (+48) 12 430 63 00
e-mail: marketing@ccpress.pl
www.ccpress.pl

Printing and binding by Bookpress.eu

Naveen Sundar Govindarajulu & Selmer Bringsjord

The Untenability of Agentless Versions of the Church-Turing Thesis

1. Introduction

In prior work (e.g., [4]), we have followed many thinkers (e.g., most recently, Peter Smith [13]) since the time of Church and Turing in taking the Church-Turing Thesis to necessarily involve *cognitive agents*. However, the fact of the matter is that some thinkers do advocate a version of this thesis that makes no reference to agents of any sort. Herein, we show that perhaps the most powerful modern defense of an agentless version of the Church-Turing Thesis, one provided by Arkoudas [2], fails. In short, Arkoudas, and any who would follow him, are impaled by either of two horns in a rather nasty dilemma: Either the agentless camp affirms a version of the thesis that makes reference to a formally specified kind of information-processing device equivalent to a Turing machine, or not. If so, the device simply enters the extensive catalogue of devices that have likewise provided evidence *for* the Church-Turing Thesis (a catalogue that e.g. contains Register machines), but is therefore certainly not a device that captures in one fell swoop the intuitive concept of effective computation (or algorithmic computation, etc.) that this thesis identifies with Turing-machine computation. If not, then these remarkably generic devices must be at least be characterized with *some* rigor—but that rigor inevitably includes reference to cognitive agents.

The plan for the present paper is straightforward: In the next section, 2, we briefly clarify the concept of agenthood in play. We then (§3) recount Bringsjord's/our version of the Church-Turing Thesis, which has been specified in numerous publications (e.g., [3, 4]), and accept as equivalent this version and a version advocated by Copeland, and opposed by Arkoudas. Next (§4), we briefly explicate the thesis that Arkoudas identifies with the Church-Turing Thesis, one supplied by Gandy. We then (§5) proceed to give our argument by dilemma for our main claim (viz., that the Church-Turing Thesis must make reference to cognitive agents). Finally, in section 6, we briefly note one immediate consequence of our results herein: viz., no purported proof of an agentless proposition is a proof of the Church-Turing Thesis, which is rather unfortunate news for Dershowitz & Gurevich [6]. We end by anticipating the next phase in our research program: a machine-verified formal proof of our agent-based version of the Church-Turing Thesis.

2. Agents *Simpliciter* vs. Cognitive Agents

Notice that we speak of *cognitive* agents. The reason is that in AI and parts of computer science, agents can lack an ability to perceive, act, intend, understand conditionals, etcetera; that is, agents in such fields can lack the powers that have been associated with Turing-level computation since the time of Post, Turing, and Church. For instance, in [12], the square-root function, if implemented in, say, Common Lisp, counts as an agent. But surely that's a very impoverished agent! When Turing spoke of "computists" using paper and pencil to compute, and Post "workers," they surely had in mind agents that could perceive objects in the real world, and act accordingly. Our point at the moment isn't that the concept of a cognitive agent must be employed in an accurate specification of the Church-Turing Thesis. Rather, our point is simply that whether or not agents are central to this thesis, the *kind* of agents that are relevant are certainly cognitive in nature.

3. Our CTT (and Copeland's CTT) as the Church-Turing Thesis

For us, at the heart of the Church-Turing Thesis stands the notion of an *algorithm*, characterized traditionally as a finite and completely specified step-by-step and initial-step-to-concluding-step procedure for solving an entire class of problems. An algorithm is prescribed in advance and does not depend upon any physical or random factors. While different words are used in different textbooks, the previous two sentences constitute an adequate distillative description of the concept in question. A function $f : A \rightarrow B$ is then called *effectively computable* iff there exists an algorithm that an idealized cognitive agent can follow in order to compute the value $f(a)$ for any given $a \in A$.[1] It is crucial to note the central role of the "idealized cognitive agent" in this characterization of effective computation. The centrality of such a creature makes effective computation an intrinsically *cognitive* concept. Without loss of generality, we can restrict attention to so-called *number-theoretic functions*, i.e., functions that take N to N (where N is the set of natural numbers). As most readers will well know, the justification for this restriction is a technique known as *arithmetization*. Using ideas made popular by Gödel, one can devise encoding and decoding algorithms that will represent any finite mathematical structure (e.g., a graph, a context-free grammar, a formula of second-order logic, a Java program, etc.) by a unique natural number. By using such a scheme, a function from, say, Java programs to graphs, can be faithfully (and *effectively*) represented by a function from N to N. Similar techniques can be used to represent a function of multiple arguments by a single-argument function.

The notion of an effectively computable function is at least seemingly informal, since it is based on the somewhat vague concept of an algorithm, and on the not-exactly-transparent concept of a cogni-

[1] We use the generic term 'cognitive agent,' or sometimes simply 'agent.' As we noted above, Turing [14] spoke of "computists" and Post [11] of "workers," humans whose sole job was to slavishly follow explicit, excruciatingly simple instructions.

tive agent. For us, the Church-Turing Thesis also involves a more formal notion, that of a *Turing-computable* function. A (total) function $f : N \to N$, recall, is Turing-computable iff there exists a Turing machine which, starting with n on its tape (perhaps represented by n |s), leaves $f(n)$ on its tape after processing, for any $n \in N$. (The details of the processing are harmlessly left aside for now; see, e.g., Lewis & Papadimitriou [9] for a thorough development.) Given this definition, we take the Church-Turing Thesis to amount to:

> **CTT** A function $f : N \to N$ is effectively computable if and only if it is Turing-computable.

Our presentation above follows the original character of the CTT in that it does not analyze what it is for a function to be effectively computable. Copeland [5] provides at least the start of such analysis that is in line with our version of the thesis, and by doing so gives us what Arkoudas labels **CTTATC** (labeled 'CTT' by Copeland):

> **CTTATC** A function f can be computed by an idealized human clerk working by rote with pencil and paper if and only if f is Turing-computable.

It is this thesis and similar agent-oriented propositions that Arkoudas claims are not acceptable expressions of the Church-Turing Thesis and so can be dismissed. So, what does a good *non*-agentless thesis look like?

4. Gandy's Thesis as the Church-Turing Thesis

Arkoudas holds that "Gandy's Thesis" (**GT**) is the correct, and indeed orthodox, version of the Church-Turing Thesis; **GT** makes no reference to agents, computists, workers, clerks, and so on. Here's the thesis:[2]

[2] So as to bring Arkoudas's discussion into conformity with the fact that the Church-Turing Thesis is a biconditional, we add the uncontroversial direction (right-to-left) to what Arkoudas takes **GT** to be, which of course yields a biconditional structure.

GT f is DDM-computable iff f is Turing-computable.

Note that here 'DDM' is short for 'digital deterministic machine,' and therefore Gandy is focusing on a key attribute for an information-processing machine or device: viz., that of being *digital-deterministic*. It will be convenient later if we on occasion simply write 'DD' for this attribute.

5. The Fatal Dilemma Facing Proponents of Agentless Theses

We begin by giving the structure of the argument which establishes that **GT** isn't a tenable agentless version of the Church-Turing Thesis. The reasoning conforms to proof-by-cases; there are two; hence the argument, as we have noted, is by dilemma.

Suppose on the one hand that DDMs are information-processing machines whose level of detail is comparable to Register machines, Turing machines, and so on. Then **GT** isn't a viable candidate for the Church-Turing Thesis. This is easy to see: Suppose for *reductio* that **GT** *is* viable in this sense. Then it follows directly that *all* versions of the Church-Turing Thesis have the form

$$\mathcal{L} \text{ iff } \mathcal{T},$$

where the left ($\mathcal{L}$) side of the biconditional ascribes a formalized property of the form $\mathcal{M}$-computable to the function f that is being universally quantified over, and where $\mathcal{T}$ simply stands for the assertion that f is Turing-computable. (In the case of **GT**, of course, DDM = $\mathcal{M}$.) But this reduces to absurdity, because *all* the biconditionals that have through the decades been established by outright proof (e.g., a function f is Register-machine-computable iff it's Turing-computable) would each suddenly quite literally become identified with the Church-Turing Thesis itself, which is absurd.[3] Pictorially put,

[3] This is not to imply that some particular formalized Turing-level machines don't do a better job than others in being congenial to the kinds of problem-solving ac-

all these biconditionals simply constitute points in what can be fairly called the "humble circle of Turing-level computation" (see Figure 1). But what is needed isn't a point *in* the circle, but rather a characterization of the circle itself!

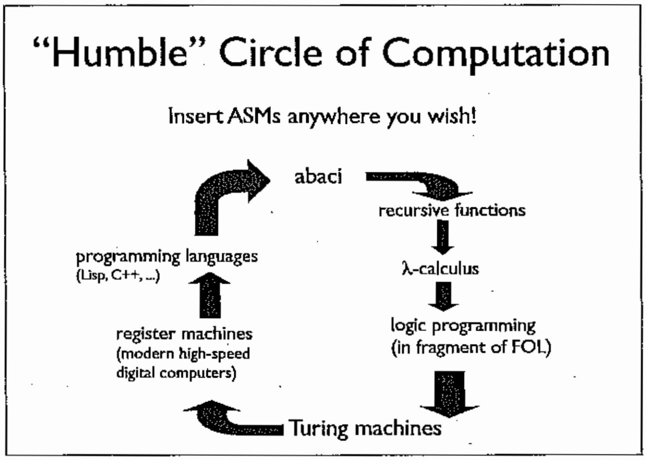

Figure 1: The Humble Circle of Turing-Level Computation. (ASMs are 'abstract state machines,' which are characterized by D&M (see section 6).)

Now suppose on the other hand that DDMs are generic information-processing machines intended to somehow capture information-processing of the *general* sort that is often labeled as "effectively computable" or—as in the case of Gandy [7] himself—"algorithmically computable."[4] Since, again, we aren't doing poetry, DDMs will satisfy this role by virtue of the attribute DD, in explicated form. But the explication of DD makes reference to cognitive agents; hence the second case is also closed.

Of course, at this point, we are simply asserting without argument that the explication of DD makes such reference. This needs to be shown; we supply the needed argument now.

---

tivities that *human* cognitive agents engage in. We happen to believe in fact that our modification of so-called KU (Kolmogorov-Uspenskii) machines [8] are in this regard quite nice (e.g., see [4])—and clearly in this regard Smith [13] is like-minded.

[4] Other phrases have of course been used.

Irrepressible Agenthood in GT

Arkoudas [2] provides an insightful explication of DD. Specifically, he presents and affirms a principle that expresses a necessary condition on DD. The principle is that if a machine has the property of being DD, it must have *systematic predictability*. We read:

> The [necessary] condition is this: systematic predictability of observable behavior. Any class $\mathcal{C}$ of deterministic digital computers ought to be amenable to systematic analysis. **We should be able to construct** mathematical idealizations of the elements of $\mathcal{C}$ such that given *any* mathematical description of a machine $C \in \mathcal{C}$ and any description of an appropriate input x, **we are able to** *predict the course of the execution of C on input x*. That is, **we should be able to make precise statements** about what will happen at any given point in the future once **we start running** C on x. In general, of course, prediction is an essential aspect of all forms of science and engineering: **We construct** a mathematical model of a class of systems (e.g., pendulums or airplanes), usually by solving an appropriate system of differential equations, and use it to make predictions about the future behavior of any particular system, given some initial conditions. It is my contention that if no such mathematical theory is available and no such predictions can be made systematically, then we are not dealing with deterministic computing machines in any reasonable sense of the term. Loosely put, if the observable behavior of a device is inherently unpredictable, then that device is not a determimistic digital computer. (Arkoudas [2, p. 5]; bolded text due to us; that emphasis exploited momentarily)

Arkoudas provides here all that we need to wrap up our argument. To see this, we have only to attend to the bolded text in the above quote. Doing so immediately reveals that systematic predictability is explicated by reference to the cognition of human agents. After all, when Arkoudas says "... we should ..." and "... we are ..." and "We construct ..." what does this pivotal 'we' refer to? Obviously the reference is to human beings, since his readers are expected to be members of this species.

One might attempt to counter by insisting that the reference to humans in the above characterization of systematic predictability is merely for convenience, and that such reference can be supplanted in favor of reference to machines. This attempt fails, as one has to countenance the ascription to these machines the powers that are part and parcel of systematic predictability—but such powers instantly serve to classify these machines as cognitive agents!

We have allowed that explication can be informal (but must be at least rigorous to a significant degree). From the standpoint of formal logic, what does the explication provided here by Arkoudas imply? What he says can be formalized naturally by (and as far as we can see, *only* by) employing a marriage of quantified conditional logic and a robust array of intensional operators for relevant propositional attitudes. To see this, note that on a formal account, the systematic predictability of a given machine/device would consist in it being the case that:

W were the device in such-and-such a state, and were it the case that this state was perceived and understood by a cognitive agent, that agent would be able to predict at least some future states of the device.

Obviously, **W** is a subjunctive conditional; therefore conditional logic is called for. And in addition, something like the cognitive event calculus $\mathcal{CEC}$ is needed to formally capture the "psychology" side of **W**. The cognitive event calculus is a first-order modal logic that incorporates the first-order event calculus and has operators for representing propositional attitudes of agents; see [1] for an introduction to the $\mathcal{CEC}$. We indicate the direction for how to render **W** rigorous by presenting an *ad hoc* formalization below using a micro logic $\mathcal{L}$ that, along the lines of the $\mathcal{CEC}$, includes sorted first-order modal logic.

$\mathcal{L}$ has sorts $\langle\mathsf{Device}, \mathsf{State}, \mathsf{Time}, \mathsf{Agent}\rangle$, predicate symbols $\langle In, < \rangle$, and operators $\langle Perceives, Understands, Predicts\rangle$. In our intended interpretation, the sorts correspond to devices, device states, time points, and cognitive agents, respectively. The predicate symbol In represents that a device is in some state at some time point. The symbol $<$ represents some natural temporal ordering. The agent-and time-indexed modal operators $\langle Perceives, Understands, Predicts\rangle$, respectively, capture cognitive agents perceiving, understanding and predicting states-of-affairs represented by propositions. The logic also has a quantifier τ for representing the natural-language quantifier *"the."* $\mathcal{L}$ also has a connective $>$ to represent subjunctive conditionals in natural language. The surface-logical structure of **W** should then be of the form[5]:

$$\tau.d\!:\!\mathsf{Device}\ \exists.\sigma\!:\!\mathsf{State}\ \exists.\ t\!:\!\mathsf{Time}\ \Big(In(d,\sigma,t)$$
$$\wedge\exists.a\!:\!\mathsf{Agent}\ Perceives(a,t,In(d,\sigma,t)) \wedge Understands(a,t,In(d,\sigma,t))\Big)$$
$$>$$
$$\exists.\sigma'\!:\!\mathsf{State}\ \exists.t't''\!:\!\mathsf{Time}\ Predicts(a,t',In(d,\sigma',t'')) \wedge (t < t' < t")$$

The fragment $In(d,\sigma,t)$ captures that the device is in some initial state. The agent's combined initial cognitive state is captured by $Perceives(a,t,In(d,\sigma,t)) \wedge Understands(a,t,In(d,\sigma,t))$. The consequent captures what it means for the cognitive agent to predict some future state of the device. We can see that there is a quite significant formalization project facing us, one that requires *sustained* explication of systematic predictability in terms of cognitive agents. This project requires us to analyze, at a minimum, what it means for a cognitive agent to perceive, understand, and predict states-of-affairs (a "physics" of abstract cognition). That the prediction should follow or entail from perception and understanding of a previous state of affairs is necessitated by the subjunctive conditional. The semantics of the subjunctive conditional should in turn take into account, at the least, the laws governing the cognition of the agent.[6] Absent such a formalization of **W**, any formalization of effective computation is either not rigorous enough or simply a formalization of a point in the humble circle of computation.

[5] Using the syntax $Quantifier.\ variable\!:\!sort$ to represent sorted quantification.
[6] Please consult [10] for why a proper semantics of subjunctives needs to have this character.

6. Upshot; The Future

One immediate upshot of the result produced above is that any purported proof of the Church-Turing Thesis that at best establishes an agentless proposition is in fact *not* a proof of this thesis. A serious and non-trivial formalization of agenthood is required to elevate any proof attempt from the humble circle into the realm of CTT. Therefore, for example, contrary to what they claim, Dershowitz & Gurevich [6] do not (informally) prove the Church-Turing Thesis. They declare postulates intended to characterize what they believe to be effective computation, to wit:

Postulate I (Sequential time). An algorithm is a state-transition system. Its transitions are partial functions.

Postulate II (Abstract state). States are structures, sharing the same fixed, finite vocabulary. States and initial states are closed under isomorphism. Transitions preserve the domain, and transitions and isomorphisms commute.

Postulate III (Bounded exploration). Transitions are determined by a fixed "glossary" of "critical" terms. That is, there exists some finite set of (variable-free) terms over the vocabulary of the states, such that the states that agree on the values of these glossary terms, also agree on all next-step state changes.

Postulate IV (Arithmetical States). Initial states are arithmetical and blank. Up to isomorphism, all initial states share the same static operations, and there is exactly one initial state for any given input values.

The thesis they claim to prove is:

> Theorem 4.5. A numeric function is partial-recursive if and only if it is computable by an arithmetical abstract state machine.

As can be seen, the postulates and purportedly proved thesis lack allusion to (cognitive) agenthood, let alone an analysis of this concept; this deflates their project to simply proving equivalence between points in the humble circle.[7]

Our next step will be to produce the first machine-verified formal proof of an agent-based version of the Church-Turing Thesis as **CTT**. This proof will exploit a logic based on the $\mathcal{CEC}$ that has as its foundation a "physics" of abstract cognition.

Bibliography

[1] K. Arkoudas, S. Bringsjord, *Toward formalizing common-sense psychology: An analysis of the false-belief task*, [in:] *PRICAI 2008: Trends in Artificial Intelligence*, eds. T.B. Ho, Z.-H. Zhou, Springer-Verlag, Berlin 2008, pp. 17–29.

[2] K. Arkoudas, *Computation, Hypercomputation, and Physical Science*, "Journal of Applied Logic", vol. 8 (2008), pp. 461–475.

[3] S. Bringsjord, K. Arkoudas, *On the Provability, Veracity, and AI-Relevance of the Church-Turing Thesis*, [in:] *Church's Thesis After 70 Years*, eds. A. Olszewski, J. Woleński, R. Janusz, Ontos Verlag, Berlin 2006, pp. 66–118.

[4] S. Bringsjord, N.S. Govindarajulu, *In Defense of the Unprovability of the Church-Turing Thesis*, "International Journal of Unconventional Computing", vol. 6 (2011), pp. 353–374.

[5] B.J. Copeland, *Computation*, [in:] *The Blackwell Guide to the Philosophy of Computing and Information*, ed. Luciano Floridi, Blackwell, Oxford 2003, pp. 3–17.

[6] N. Dershowitz, Y. Gurevich, *A Natural Axiomatization of Computability and Proof of Church's Thesis*, "The Bulletin of Symbolic Logic", vol. 14 (2008), pp. 299–350.

[7] There are additional problems with D&G's attempt, but we don't discuss them here. For instance, their effort is complicated by the fact that syntactic entities (such as terms and vocabularies) and semantic entities (such as structures) are indiscriminately glued together in definiug their abstract state machines (which, as we have noted, are in turn intended to capture effective computation).

[7] R. Gandy, *Church's Thesis and Principles for Mechanisms*, [in:] *The Kleene Symposium*, eds. J. Barwise, H. Kreisler, and K. Kunen, North-Holland, Amsterdam 1980, pp. 123–148.

[8] A.N. Kolmogorov, V.A. Uspenskii, *On the Definition of an Algorithm*, "Uspekhi Matematicheskikh Nauk", vol. 13 (1958), pp. 3–28.

[9] H.R. Lewis, C.H. Papadimitriou, *Elements of the Theory of Computation*, Prentice Hall, Upper Saddle River, New Jersey 1997.

[10] D. Nute, *Conditional logic*, [in:] *Handbook of Philosophical Logic Volume II: Extensions of Classical Logic*, eds. D. Gabay, F. Guenthner, Reidel, Dordrecht 1984, pp 387–439.

[11] E. Post, *Recursively enumerable sets of positive integers and their decision problems*, "Bulletin of the American Mathematical Society", vol. 50 (1944), pp. 284–316.

[12] S. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, Upper Saddle River, New Jersey 2009. Third edition.

[13] P. Smith, *An Introduction to Gödel's Theorems*, Cambridge University Press, Cambridge 2007.

[14] A.M. Turing, *On Computable Numbers with Applications to the Entscheidungsproblem*, "Proceedings of the London Mathematical Society", vol. 42 (1937), pp. 230–265.