**This is an Accepted Manuscript of an article published by Taylor & Francis in *Philosophical Psychology* on 28.04.2016, available online: http://www.tandfonline.com/10.1080/09515089.2016.1150450**

**Physical Computation: A Mechanistic Account**

Gualtiero Piccinini

Oxford: OUP, 2015

336 pages, ISBN: 978-0-19-965885-5 (hbk); £35

*Physical Computation* is the summation of Piccinini’s work on computation and mechanistic explanation over the past decade. It draws together material from papers published during that time, but also provides additional clarifications and restructuring that make this the definitive presentation of his mechanistic account of physical computation. This review will first give a brief summary of the account that Piccinini defends, followed by a chapter-by-chapter overview of the book, before finally discussing one aspect of the account in more critical detail.

 For those who are unfamiliar with the account, here is roughly how Piccinini proposes we should characterise physical computation. A computing mechanism consists of a series of physical components individuated in terms of their computational function – for example, a processor is a component that performs systematic transformations on strings of digits, and an input device is a component that transforms external stimuli into digits. These components are in turn composed of further subcomponents that eventually bottom out with “primitive computing components (paradigmatically, logic gates)” (119)[[1]](#footnote-1), which cannot be decomposed any further without leaving the realm of computation (although they are amenable to further non-computational mechanistic decomposition). Thus the notion of physical computation is cashed out mechanistically, without having to appeal to any notion of representation or content. The account is also able to successfully meet each of the six desiderata that Piccinini establishes in the first chapter of the book, as he demonstrates at the end of chapter 7.

 The book opens with an outline of Piccinini’s overall project, including a brief history of the philosophical understanding of computation (4-8), an initial definition of his mechanistic account of computation (10), and perhaps most importantly, a list of the six desiderata that he thinks any account of physical computation must meet (11-15). These desiderata are *objectivity*, *explanation*, *the right things compute*, *the wrong things don’t compute*, *miscomputation is explained*, and finally that an adequate *taxonomy* can be given. Each is fairly self-explanatory, but they are important to keep in mind when reading the book, as they guide the direction and flavour of Piccinini’s account.

 The next three chapters give an overview and critique of classical accounts of computation, including mapping accounts (chapter 2), semantic accounts (chapter 3), and a whole chapter dedicated to discussing (and dismissing) pancomputationalism (chapter 4). These chapters will be useful to anyone new to the topic, or anyone wanting to refresh their memory, and Piccinini’s presentation of these issues is clear and insightful.

 The bulk of the book, from chapter 5 through to chapter 13, is dedicated to a thorough presentation of Piccinini’s mechanistic account of physical computation. This summarises and expands on Piccinini’s previous work, outlining the relationship between functional analysis and mechanistic explanation (chapter 5), presenting an account of functional mechanisms (chapter 6), describing the basic structure of a computing mechanism (chapter 7), and providing a taxonomy of the different kinds of computing mechanisms that exist (chapter 10-13). There is new content presented in chapters 8 and 9 that describes in more detail the kinds of components that one can expect to find in a typical computing mechanism. Chapter 8 describes how primitive computing components perform basic logical transformations, whilst chapter 9 describes how complex components combine these basic transformations in order to perform more powerful computations.

 The final section of the book explores a couple of classical issues in light of the mechanistic account of physical computation: information processing and the Church-Turing thesis. Chapter 14 brings some much-needed clarity to our use of the term information, most notably by reminding us of the distinction between (non-semantic) Shannon information and various forms of semantic information. These two uses of the term ‘information’ are easily conflated, and keeping them separate, Piccinini contends, is crucial to our understanding of computation. In the second half of the chapter Piccinini explicates the complicated relationship between computation and information. He concludes that information processing (of any kind) will always co-occur with computation in the generic sense, as both are defined in terms of the processing of medium-independent vehicles. However, computations need not necessarily carry semantic information, even if in practice most do. There is nothing especially controversial here, but it is an important topic to clear up.

 Chapters 15 and 16 articulate and resolve some of issues associated with formulating a physical version of the Church-Turing thesis; that is to say, the thesis that all functions performed by physically realised computing mechanisms are Turing-computable, and that a physical Turing machine could perform any of these functions (given sufficient time and/or memory). Piccinini rejects a “bold” version of this thesis (“Any physical process is Turing computable”), but argues that we should accept a “modest” version (“Any function that is physically computable is Turing computable”). The material in these final two chapters is very technical and does not always seem entirely relevant to the thesis of the book as a whole. It could nonetheless prove to be a useful resource for anyone involved in these specific debates.

 All of this will be familiar to those who have been following the progress of Piccinini’s account over the years. There are only a few points at which he diverges from what he has written previously, and for the most part these divergences are relatively minor. The only major change is that Piccinini now seems to concede more than he did previously to those who defend the role of semantic content in our understanding of computation. For example, in chapter 3 he writes, “computation may be sensitive to semantic properties, at least in some cases, while having a wholly non-semantic nature” (31-2). This is a notable softening of his earlier position, which maintained that computation was completely autonomous from semantic content (although not necessarily vice versa). He does add the caveat that this applies only to “certain narrow meanings”, such as the difference between *add* and *subtract*, but even this is enough to provide a much stronger role for semantic content than he has previously allowed.

 This trend is even more apparent in his presentation of mechanistic explanation, in which he relies upon a seemingly robust notion of teleological functions. Given that he wants to do away with the semantic theory of computation, this seems like an odd move to make, as once we have teleological functions we are not far off having a full-blown teleosemantic theory of representation. Even if his account of teleological functions is successful, he will have to demonstrate how it differs from teleosemantics, or else risk undermining his own rejection of the semantic theory of computation. I found this to be the weakest section of the book, and as such it deserves some greater scrutiny.

 Piccinini claims to offer “an ontologically serious account of functional mechanisms” (100), one that is able to “identify non-teleological truthmakers for teleological claims” (103). He rejects prior attempts, including those that he thinks either make functions causally impotent (etiological and selectionist accounts), or make functions relative to explanatory interest, neither of which he is satisfied with. His own account of teleological functions is grounded “directly in the special organismic goals of survival and inclusive fitness” (106). A function is teleological if it “is a stable contribution by a trait […] of organisms belonging to a biological population to an objective goal of those organisms” (108), where an objective goal is one that contributes to survival and/or inclusive fitness. For example, the heart’s proper function is to pump blood, rather than to produce noise, because this function contributes to the survival of the organism, and in the long term to its inclusive fitness.

 It is not clear how successful an account of this kind will be in avoiding the kind of concerns that Piccinini himself expresses with regard to the semantic account of computation. One reason that he gives for rejecting the semantic account is that it requires a non-semantic computational structure before it can get off the ground, at which point it no longer seems to contribute much to our understanding of computation (33-4). However, in appealing to survival and inclusive fitness as objective goals of organisms Piccinini seems to be committing a similar error. The fact that the mechanistic structure of an organism contributes to its survival or inclusive fitness does not by itself contribute anything to our understanding of those mechanisms. Once we are able to explain how the mechanisms function, we can tell a further story about how they contribute to survival and inclusive fitness, but appealing to this story as a part of the initial mechanistic explanation seems to introduce a potentially vicious circularity of the very same kind that Piccinini warns us against when discussing the semantic account. The problem could be avoided if the attribution of teleological functions was merely an explanatory heuristic, rather than an intrinsic part of the explanation itself, but this seems little different to the explanatory interest account that Piccinini explicitly rejects.

 As noted above, there is a further issue that will arise even if this account is successful. Once we have objective teleological functions, it is relatively simple to derive a teleosemantic account of representations, for example by positing mechanisms whose function is to represent. At this point the semantic account of computation is no longer reliant on a non-representational computational structure, as it can legitimately appeal to representational mechanisms as a basic component of computational systems. Piccinini’s argument becomes self-defeating, as by invoking objective teleology he removes one of the primary motivations for giving a non-representational account of computation in the first place. There may be other motivations, and we could retain his account even if there weren’t, but it would no longer be the only game in town.

 None of this should be taken as a major criticism of the book as a whole. I have every hope that this issue can be resolved, although as it stands it threatens to undermine the mechanistic basis of Piccinini’s account. Despite this threat, I remain convinced that the mechanistic account of physical computation is the best that we currently have, and I look forward to seeing it developed further in the future. *Physical Computation* is eminently readable and well presented, with a clear structure and helpful introduction. Whilst it does occasionally slip into more technical language, this is unavoidable considering the subject area, and it should be accessible to an educated non-expert. It provides a thorough (if understandably biased) introduction to the philosophical issues associated with computation in the physical sense, and would serve as a good basis for a postgraduate or high-level undergraduate course on the subject. Piccinini delivers a comprehensive summary of previous work on physical computation, alongside the definitive presentation of his mechanistic account, and I have no doubt that this book will become a valuable resource for future work on the topic.

1. All references are to Piccinini (2015). [↑](#footnote-ref-1)