On Two Different Kinds of Computational Indeterminacy*

Philippos Papayannopoulos[†], Nir Fresco[‡] and Oron Shagrir[§]

[†]*Department of Philosophy, The Hebrew University of Jerusalem.*
[‡]*Department of Brain and Cognitive Sciences, Department of Philosophy, Ben-Gurion University of the Negev.*
[§]*Department of Philosophy, Department of Cognitive and Brain Sciences, The Hebrew University of Jerusalem.*

Abstract

It is often indeterminate what function a given computational system computes. This phenomenon has been referred to as "computational indeterminacy" or "multiplicity of computations". In this paper, we argue that what has typically been considered and referred to as the (unique) challenge of computational indeterminacy in fact subsumes two distinct phenomena, which are typically bundled together and should be teased apart. One kind of indeterminacy concerns a functional (or formal) characterization of the system's relevant behavior (briefly: how its physical states are grouped together and corresponded to abstract states). Another kind concerns the manner in which the abstract (or computational) states are interpreted (briefly: what function the system computes). We discuss the similarities and differences between the two kinds of computational indeterminacy, their implications for certain accounts of "computational individuation" in the literature, and their relevance to different levels of description within the computational system. We also examine the interrelationships between our proposed accounts of the two kinds of indeterminacy and the main accounts of "computational implementation".

1 Introduction

A variety of fields, from physics and chemistry to the cognitive and brain sciences, have begun to import the language of computation and information processing to describe their studied

*This is a preprint of an article that has been accepted for publication in *The Monist*. Please cite the published version that would appear here: https://academic.oup.com/monist

systems. Interpreting certain physical systems as carrying out specific computations can provide additional insights into a system's behavior and organization. However, characterizing a system computationally is generally a challenging task, even when the system's physical behavior is sufficiently known. This is because there may be more than one computational profile consistent with the same, underlying physical behavior. In other words, some systems exhibit what is often described as *computational indeterminacy*: a detailed description of the physical dynamics of the (closed) system does not suffice to pin down a unique computational identity for it. To determine such an identity, typically some *external* knowledge and hypotheses about the system are invoked as well; perhaps parameters about its environment, its computational goals, the semantic content of its computational states or others. This is a well-known phenomenon, of course, and has been extensively discussed in the literature, under various different guises and names.[1]

The aim of this paper is to show that what is typically considered and referred to as *the* phenomenon of computational indeterminacy, in fact subsumes two separate phenomena that should be distinguished. We present the two phenomena as two different kinds of indeterminacy and attempt to show that their varied nature is owing to different inter-level relations pertaining to the process of computationally individuating the system. One kind of (computational) indeterminacy arises at the level of formally or functionally characterizing the system, and another kind arises at the level of interpreting what the computation is about (what is computed or, in other terms, what the contents of the computational states are).

Computational individuation intertwines with computational implementation, in a way that makes it difficult to give a nuanced account of the former without taking into consideration accounts of the latter. In what follows, however, we will discuss the different kinds of (computational) indeterminacy without presupposing any specific, nuanced views of 'implementation' and 'computation' in the philosophical literature. We will also examine how our proposed accounts of the two kinds of indeterminacy relate to existing views on 'implementation' and 'computation', but take no specific stand on which view is preferable.[2]

Section 2 discusses the two kinds of indeterminacy in detail and examines aspects of the relations between them and computational implementation. Section 3 explores the interrelationships between the two indeterminacies, computational individuation and levels of organization

[1]Some examples of such names include 'simultaneous implementation' (Shagrir 2001, 2020, Fresco 2015, Dewhurst 2018), 'the ambiguity of representation' (Maroney and Timpson, 2018), 'indeterminacy of computation' (Fresco et al., 2021), 'underdetermination of computation' (Duwell, 2018), 'multiple-computations theorem' (Hemmo and Shenker, 2019), 'multiplicity of computations', and various others.

[2]In doing so, we may be seen as adopting the point of view of a scientist who employs computational characterizations for her studied systems and who may or may not have a particular theory of implementation in mind. But all the same, the scientist's computational hypotheses bear important epistemic (and perhaps ontological) weight, for they underpin an understanding of the system that is complementary to its physical description and yet not reducible to it. Such non-reducibility arises, partly, exactly because the physical dynamics alone does not suffice to uniquely determine some computational description.

in computational systems. Section 4 concludes the paper.

2 Two kinds of (computational) indeterminacy

We start by briefly characterizing the two kinds of indeterminacy. Suppose that we are trying to specify the computational profile of a certain physical system, such as a neuronal network or an artifact, with known physical behavior. A first challenge concerns how to carve up the physical state space of the system in order to correspond physical states to abstract ones. Such a correspondence underlies descriptions of the functional organization of the system. Differently put, the question concerns which physical processes should be grouped together into parts of the same abstract/functional state. As we show in the next section, there are systems susceptible to varied ways of grouping physical states together, such that there is no inherent reason for preferring one way to another; consequently, some external restrictions may be required to functionally characterize the system (e.g., contextual, mechanistic, semantic or other).

Once an abstract/functional organization is specified, there exists a second kind of indeterminacy that pertains to computational individuation. This concerns specifying what is computed by the system; that is, interpreting either the content of the abstract/functional states or what the physical states represent.[3] For example, a given functional profile of a system can give rise to more than one semantic interpretation of it. To see this, consider a physical system whose functional profile is consistent with implementing some logic gate (say, an AND-gate). For certain gates, by swapping the logic values that we take a certain voltage range to represent, we obtain a different (but related) Boolean function, which is called the *dual* of the first gate (in our case, an OR-gate). The choice between the two interpretations, then, is indeterminate, unless some external restrictions are again provided.

Let us now turn to discuss each kind of indeterminacy in more detail and examine some concrete examples.

2.1 Functional indeterminacy

Consider a physical system $\mathcal{P}$, which is a tri-stable flip-detector. $\mathcal{P}$ works as follows: it emits between 7–10 volts if it receives voltages between 7–10 volts from exactly one input channel, it emits 0–3 volts if it receives between 0–3 volts from each input channel, and it emits between 4–6 volts in any other case. The behavior of $\mathcal{P}$ is summarized in table 1:

The next step is to abstract from these physical voltages and to present the mapping in terms of three different types or sets, where A, B and C stand for variables of sets; see table 2, for

[3] We explain why we allow here a disjunction about what is interpreted (the physical or the abstract states) in the next section.

Input 1 (V)	**Input 2 (V)**	**Output (V)**
7–10	7–10	4–6
7–10	4–6	7–10
7–10	0–3	7–10
4–6	7–10	7–10
4–6	4–6	4–6
4–6	0–3	4–6
0–3	7–10	7–10
0–3	4–6	4–6
0–3	0–3	0–3

Table 1: Physical gate $\mathcal{P}$: The gate maps voltages from two input channels, *Input 1* and *Input 2*, into one *Output* channel.

example.

Input 1	**Input 2**	**Output**
A	A	B
A	B	A
A	C	A
B	A	A
B	B	B
B	C	B
C	A	A
C	B	B
C	C	C

Table 2: The functional organization of $\mathcal{P}$. The gate maps abstract types from two input channels, *Input 1* and *Input 2*, into one *Output* channel.

What is the difference between tables 1 and 2? The important difference is that table 2 abstracts from the physical properties (voltages) of table 1. Table 2 says that we have an input-output mapping between three different types, but it is blind to the type of the physical properties

(e.g., voltages). The terms A, B, and C can be seen as variables of sets, any sets. Table 1 signifies one way to implement (or realize) this functional mapping, namely, in voltages. But the mapping of table 2 could be implemented ("multiply realized") by different voltages and even by other physical properties. In this respect, we can say that table 2 describes the *functional organization* (Putnam, 1967), or the *medium-independent structure* (Haugeland 1978; Piccinini 2015), or the *organizationally invariant* or *abstract causal structure* (Chalmers, 2011) of $\mathcal{P}$.

Now, there are further alternatives in grouping the voltages. For example, we could have a coarser-grained grouping under which we put {0–3 volts; 4–6 volts} in one type and {7–10 volts} in another type. Abstracting from the physical properties, as before, we get the mapping of table 3.

Input 1	**Input 2**	**Output**
A	A	B
A	B	A
B	A	A
B	B	B

Table 3: A coarser-grained abstraction from the physical gate $\mathcal{P}$. The gate maps abstract types, A and B.

We could also put {0–3 volts} in one type, B', and {4–6 volts; 7–10 volts} in another type, A'. Abstracting from the physical properties, we get table 4.

Input 1	**Input 2**	**Output**
A′	A′	A′
A′	B′	A′
B′	A′	A′
B′	B′	B′

Table 4: Yet another coarser-grained abstraction from the physical gate $\mathcal{P}$. The gate maps abstract types, A' and B'.

What we have here is *functional indeterminacy*: the physical system $\mathcal{P}$ can be seen as having (at least) three different functional profiles,[4]

presented in tables 2-4 respectively.[5]

[4]There are, of course, other combinations theoretically available, such as grouping {0–3 volts; 7–10 volts} as one type and {4–6 volts} as another, or even *the identity* grouping, where all voltages are taken as one type, {0–3 volts; 4–6 volts; 7–10 volts}.

[5]See Hemmo and Shenker (2019, 95-96) for a variant example of a physical system that exhibits the same kind of indeterminacy.

Let us be more precise about this claim. First, the functional (abstract) structures in our example are simple input-output mappings. In general, the functional (abstract) structure may also include the mediating (algorithmic) mechanism. Adding algorithms to the equation, however, does not minimize the potential for indeterminacy, since different input-output functions are always associated with different algorithms. In other words, there is always at least as much indeterminacy in algorithmic mappings as there is in input-output mappings, since a given input-output mapping can be catered for by more than one algorithmic mapping. Second, the example focuses on set-theoretic input-output functions, but the same goes for other abstract structures.[6] In general, the phenomenon here is that a physical system can correspond to several different formal structures.

Third, there exist different sorts of functional indeterminacy. Underlying all of these is that we group together physical features of the system and then abstract from them. On one subspecies of functional indeterminacy, we can group physical properties that occur in different times and spaces within the system.[7] On another subspecies of functional indeterminacy, we can group different properties of the system, in one case (say) voltages and in another case (say) temperatures, even if they occur in the same time and space. In the case of indeterminacy described here, which is perhaps the most interesting of the functional type, we group together the very same physical properties (e.g., voltages) that occur in the same time and space. In the examples above, we group the same physical properties, namely voltages, in different ways.

A mapping from the physical system $\mathcal{P}$ to each of the abstract structures above is often seen as underpinning an *implementation* (or *realization*) relation. Given the extensive discussion in the literature about the relation between implementation and indeterminacy, a question now arises: is functional indeterminacy a case of *computational* indeterminacy?[8] The answer depends crucially on one's view on the former relation —i.e., between implementation and indeterminacy— and people have supplied very different answers. Putnam (1988) and Searle (1992) famously advance triviality results that, when taken to the extreme, state that every physical system implements every functional or abstract structure (formalism). On their view, this result amounts to radical computational indeterminacy; namely, that every physical system computes every Turing-computable function (*unlimited pancomputationalism*). Many others suggest that the implementation relation goes far beyond the simple homomorphic mapping assumed by Putnam and Searle. Chalmers (1996, 2011, 2012), for one, suggests that there are further, causal,

[6] For a different example that does not involve a set-theoretic input-output function, see the discussion in sec.2.2. Furthermore, there can arguably be different levels of abstraction even within the abstract realm itself, as it becomes apparent from the distinction between 'algorithms' and 'programs' (the latter being rather more concrete); see, e.g., Turner (2021) for a discussion.

[7] For example, within a large neuron we can group differently various local physical transformations at different parts (dendrites, soma, spike initiation zone, etc.) and obtain different —equally legitimate— abstract structures.

[8] An analogous question arises with respect to the second kind of indeterminacy as well; we address that issue in sec.2.2.

modal and other constraints on the implementation relation that block radical computational indeterminacy. In the example above, the different mappings satisfy the additional constraints and nonetheless lead to functional indeterminacy. Thus, Chalmers admits that there always exists computational indeterminacy to some extent, in the sense that some physical systems perform more than one computation; however, he does not think that this is an alarming problem to be resolved.

Unlike Chalmers, proponents of the mechanistic account of computation aim to single out the computational structure of the system. According to some of them, the missing ingredient in Chalmers's account of implementation/computation is a teleological function. The claim is that the computational structure of the system is a certain abstract structure that is a stable contribution to the goal of the system. Other abstract structures do not have the same contribution.[9] Coelho Mollo (2019) and Dewhurst (2018) think that the relevant structure is the basic one (table 2),[10] whereas Piccinini (2015) argues that the computational structure can vary across contexts; it depends on the kind of interaction between the system and its close environment. Fresco and Miłkowski (2019) agree with Piccinini that computation is context-dependent, but offer a different account of the nature of the system-environment interaction. Pragmatists suggest that the feature that singles out the relevant computation depends on the explanatory aims of the scientists (Egan 2012, Rescorla 2014, Matthews and Dresner 2017). Proponents of the semantic account suggest that the missing ingredient is the content of the physical states (e.g., Sprevak 2010). Shagrir (2001, 2020), who also adopts a semantic view, further distinguishes between the notions of 'implementation' and 'computation': implementation is non-semantic whereas computation is content-dependent. According to Shagrir, the system $\mathcal{P}$ implements *all* the abstract structures above (hence his use of the term 'simultaneous implementation'), but it typically computes only one of the functions, the determining factor for which is content.

Another way to describe functional indeterminacy is in terms of *computational vehicles* (Rescorla 2014, Shea 2018). For present purposes, we take computational vehicles to be abstract causal structures that are implemented in some physical substance. On this assumption then, tables 2-4 present three different *potential* computational vehicles that are implemented by $\mathcal{P}$. One debate about these vehicles is whether each one of them is also an actual vehicle (Putnam 1988, Searle 1992, Chalmers 2011) or whether only one of these structures serves as a computational vehicle in a given context (as most others think). Those who hold the latter view debate about the missing determining factor; namely, the factor that determines which of the abstract, causal structure is *the* computational structure of the system. Coelho Mollo (2019) and Dewhurst (2018) claim that it is the same basic structure (table 2) all along; semanticists claim that it is content; Piccinini

[9]The goal of the system is often seen in terms of what the system was designed to do or has been used for. The latter are further characterized in evolutionary (Millikan, 1984), learnability (Dretske, 1988), dispositional (Maley and Piccinini, 2017), or other terms.

[10]Dewhurst's proposal is couched purely in terms of 'physical structures'.

Input 1	Input 2	Output
T	T	T
T	F	T
F	T	T
F	F	F

Table 5: The physical system $\mathcal{P}$ implements the OR function when interpreting type A' (or 4–10 volts) as indicating the truth-value T and type B' (or 0–3 volts) as indicating the truth-value F.

Input 1	Input 2	Output
F	F	F
F	T	F
T	F	F
T	T	T

Table 6: The physical system $\mathcal{P}$ implements the AND function when interpreting type A' (or 4–10 volts) as indicating the truth-value F and type B' (or 0–3 volts) as indicating the truth-value T.

claims that these are contextual, yet non-semantic factors, and so on.

2.2 Interpretative indeterminacy

A different sort of indeterminacy arises from the interpretation given to the physical/abstract states of the system. Consider table 4, for example. If we interpret type A' (or 4–10 volts) as indicating the truth-value T, and we interpret type B' (or 0–3 volts) as indicating the truth-value F, the system may be said to perform the logical function OR (table 5). However, if we interpret the (same) type A' as indicating the truth-value F, and we interpret the (same) type B' as indicating the truth-value T, the system may be said to perform the logical function AND (table 6).

Note that from a functional point of view, there is no difference between tables 5 and 6. The OR/AND difference surfaces when we interpret the same physical/functional structure in different ways. We, thus, call this kind of indeterminacy *interpretative indeterminacy*. Functional and interpretative indeterminacies, then, are different in some respects. Functional indeterminacy is related to different functional (abstract) structures, whereas interpretative indeterminacy is related to different interpretations of the same functional (abstract) structure.

Interpretative indeterminacy may extend beyond Boolean functions to functions in discrete and continuous mathematics; for example, addition and multiplication over the real numbers (Fresco et al., 2021). Let us consider an example. Suppose that a physical system has a determined functional structure that is interpreted as implementing addition. That is, the set-theoretic function from the inputs to the output of the system can be expressed as an addition of the form:

$$\text{Input}_1 + \text{Input}_2 = \text{Output}$$

It is obvious that the system can be interpreted as implementing addition. Nevertheless, it is

not the only possible interpretation. Consider the case where we interpret the inputs and output through the correspondence: $x \mapsto a^x$ $(a \neq 1)$. The output is, then, interpreted as the product of the inputs, and the same system can be seen as implementing multiplication. Table 7 shows an example, based on the correspondence: $x \mapsto 2^x$.

Input 1	**Input 2**	**Output**
0	0	0
0	1	1
1	1	2
1	2	3
2	3	5
-5	4	-1

Input 1	**Input 2**	**Output**
1	1	1
1	2	2
2	2	4
2	4	8
4	8	32
1/32	16	1/2

Table 7: An example of interpretive indeterminacy. An adder (left table) can be used as a multiplier (right table) under a different interpretation of the same inputs-output relation of a given system. The two interpretations are related through the mapping $x \mapsto 2^x$ (where x the values of the left table).

This kind of indeterminacy applies to other pairs of operations as well; for example, subtraction and division, exponentiation and multiplication, root extraction and division. Interestingly, these are not only theoretical examples with no practical relevance. In fact, some of them have been exploited in practice. For instance, it is interpretative indeterminacy that has underlain the practicality and extensive use of slide rules (through the inverse mapping $x \mapsto \log_a x$). In the simplest type of slide rules for example, while the physical transformations amount to adding the inputs (corresponded to physical lengths on the instrument), the interpreted operation is multiplication (since the lengths are graduated in a logarithmic scale).

It is important to stress again that this is not another facet of *functional* indeterminacy. The different operations stem from a different semantic interpretation of the *same* physical types (e.g., of some system similar to those in tables 1 and 8) or the same functional organization of some physical system (e.g., of a kind similar to tables 2 or 9).[11]

Functional and interpretative indeterminacies can be found together in some systems. Let us consider an example. Assume a physical system $\mathcal{Q}$ that is again a tri-stable flip-detector,

[11]This disjunctive statement depends on whether one holds that the domain of the interpretation relation is directly the physical states or some intermediate abstract structure; an issue that we discuss in more detail in sec.2.2.1.

like system $\mathcal{P}$ above. The system looks like a black box to us, but suppose that by means of appropriate measurements we obtain the input/output profile that is summarized in table 8.

Input 2 (V)	**Input 2 (V)**	**Output (V)**
0.8–0.9	0.8–0.9	0.64–0.81
0.8–0.9	0.4–0.5	0.32–0.45
0.8–0.9	0–0.2	0–0.18
0.4–0.5	0.8–0.9	0.32–0.45
0.4–0.5	0.4–0.5	0.16–0.25
0.4–0.5	0–0.2	0–0.1
0–0.2	0.8–0.9	0–0.18
0–0.2	0.4–0.5	0–0.1
0–0.2	0–0.2	0–0.04

Table 8: Physical system $\mathcal{Q}$: The system maps voltages from two input channels, *Input 1* and *Input 2*, into one *Output* channel.

How can we go about characterizing system $\mathcal{Q}$ computationally? Naturally, we can proceed as before (system $\mathcal{P}$) and determine some functional organization, by abstracting from the physical voltages. Table 9 provides such a functional organization in terms of three type variables, A, B and C. Then by grouping B and C together (i.e., from 0 to 0.5V) and keeping A separate (0.5–0.9V), we get the coarser-grained mapping of table 10. Table 10 is interpretatively indeterminate, similar to tables 3 and 4. Different interpretations of A and B as True or False give rise to either an AND or an OR.

Nevertheless, the physical behavior from table 8 admits, at the same time, of a very different abstract characterization from the one given in table 9. Observe that the mapping relating the physical inputs and outputs is also consistent with a multiplication description:

$$V_{\text{out}} = V_1 \cdot V_2$$

This, then, gives rise to an alternative abstract/functional structure expressed by:[12]

$$\text{Output} = \text{Input}_1 \cdot \text{Input}_2$$

Accordingly, system $\mathcal{Q}$ can be interpreted as computing a multiplication function. But, in line with our previous example of interpretative indeterminacy in table 7, the system could also be interpreted as carrying out addition, via the correspondence $x \mapsto \log_a x$, for some $a \neq 1$.

[12]So far, this is functional indeterminacy, because it pertains to a level of different possible functional organizations.

Input 1	Input 2	Output
A	A	A
A	B	B
A	C	C
B	A	B
B	B	C
B	C	C
C	A	C
C	B	C
C	C	C

Table 9: The functional organization of $\mathcal{Q}$. The gate maps abstract types from two input channels, *Input 1* and *Input 2*, into one *Output* channel. The typing in this table is A: 0.5–0.9V, B: 0.3–0.5V, C: 0–0.3V.

Input 1	Input 2	Output
A	A	A
A	B	B
B	A	B
B	B	B

Table 10: A coarser-grained abstraction from the physical gate $\mathcal{Q}$. The gate maps abstract types, *A* and *B* based on the typing A: 0.5–0.9V, B: 0–0.5V.

It is interesting to note that functional indeterminacy in this example stems from groupings of physical properties at different levels of granularity. The functional profile of a multiplication operation arises from a more fine-grained carving of the physical state space than that of the AND/OR operation. On the other hand, the two different functional profiles of tables 3 and 4 stem from variant groupings that exist at the same level of granularity.

It is not always easy to distinguish between functional and interpretative indeterminacies. A reason for this is that performing a computation often involves both functional and interpretative relations. Consider table 3 again. If we interpret type A (or 7–10 volts) as indicating the truth-value T, and we interpret type B (or 0–6 volts) as indicating the truth-value F, the system may be said to perform the logical function XOR. The system can now be seen as computing XOR (based on table 3) and OR (based on table 4). By interchanging the truth values, T and F, the system can be seen as computing XOR and AND (based on the same tables respectively). Is this a functional or interpretative indeterminacy? The answer is that there is a functional dif-

ference here, since the functional types that underlie the two logical functions, XOR, on the one hand, and OR/AND, on the other, are different. Still, this functional indeterminacy allows for interpretative indeterminacy. The same functional type that underlies OR can be re-interpreted as AND (the same holds for a re-interpretation of the functional type that underlies XOR as NXOR). This does not mean that every re-interpretation is allowed though. As Fresco et al. (2021) note, AND and OR are dual functions in the sense that one can swap the T for F and vice versa. XOR and OR, however, are not dual functions. OR cannot be attained from XOR by switching truth-values. The XOR/OR (or XOR/AND) indeterminacy here is of the functional sort.

2.2.1 Further properties of interpretative indeterminacy

In examining interpretative indeterminacy even further, it is instructive to regard 'interpretation' *mathematically* —as a mapping relation from a domain to a range. What are exactly the domain and range of the interpretation, then? Let us consider the *domain* component first. One can think about the interpreted states as physical, abstract, or neutral (as in the Davidsonian tradition in which states and events are individuated without respect to their descriptions/properties). Choosing *physical types* —e.g., some fixed voltage range— might seem more fitting in the context of the computational neuroscientific practice, where we often take the nervous system to be a physical system whose physical magnitudes encode, carry information about, or represent certain things. Choosing *abstract types* might be more fitting in the context of logic and computability theory, where we interpret the formal structure (e.g., strings) of (say) a Turing machine as representing (say) numbers. It, thus, makes sense to think of the physical system as implementing a formal/functional structure, which, in turn can be interpreted as representing another structure. We remain neutral on this matter, here.

When considering the *range* component of the interpretation relation, things might get even more complicated. In general, a physical computing system can be interpreted as encoding (representing) anything. It can represent physical entities in its proximal or distal environment, and even within the system itself. It can be interpreted as representing entities in counterfactual scenarios (e.g., unicorns). It can be interpreted as representing entities in mathematical or logical universes, and so on. Which content is relevant to interpretative indeterminacy? One could think that any change in content would lead to interpretative indeterminacy. But, in fact, in the examples of interpretative indeterminacy that are found in the literature the ranges of the interpretations are limited to universes that are populated by formal, mathematical, logical or other abstract entities, such as numbers, sets, or truth-values (as in tables 5 and 6). Egan (1995) refers to this type of content as *mathematical content*; Shagrir (2001) calls it *formal content*.[13]

[13]Burge (1986) and others claim that specific content is an essential element of the computational structure, and hence every difference in content makes also a computational difference (but see Egan 1995, who argues against this view). We do not make this assumption. Interpretative indeterminacy is associated with the claim that a difference

What is the relation between *interpretative* indeterminacy and *computational* indeterminacy? This depends on one's views about computational individuation. Most scholars admit that interpretative indeterminacy reflects computational indeterminacy; they debate on whether these indeterminacies further reflect that computational individuation is affected by content (e.g., Sprevak 2010) or not (e.g., Fresco and Miłkowski 2019, Fresco et al.). Others, however, think that interpretative indeterminacy does not reflect computational indeterminacy. According to the latter view, the interpretative indeterminacy is at the level of (formal) semantics which has not to do with computational individuation.[14]

Are there any restrictions on the interpretation function? Consider first the case where the domain of interpretation is directly the relevant physical states or quantities. Then allowing unconstrained interpretations can give rise to *objet trouvé* computations; that is, relatively simple physical systems can be seen as solving very hard or even uncomputable problems.[15] If, on the other hand, the domain of interpretation is abstract states or (strings of) symbols, then again unconstrained interpretations can give rise to unintended ("deviant") computations; for example, relatively simple (universal) Turing machines can be seen as easily computing hard or even uncomputable problems.[16] In any case, we will not attempt here to draw a line between admissible and deviant interpretations (be they interpretations of *physical* or *abstract* states), but we will note that without such a line we get some *radical* interpretative indeterminacies.

What is the difference between *interpretation* and *implementation*? Both can be seen as a homomorphic mapping from the physical system to some target (range).[17] However, the target of implementation is restricted to formal domains, whereas the target of interpretation might also be some other, non-formal, domains, as we have already said. Another difference, on which most scholars would agree, is that interpretation and content require more than a mapping relation; it is thought that an interpretation ('representation') is the (teleological) function to have

in mathematical content makes a computational difference, and, thus, changing the mathematical content of the same functional structure will alter the computational identity of the system.

[14]Coelho Mollo, for example, claims that the "individuation by logical function ... may well rely on wide functions or semantic properties" (2018, 3492). Dewhurst claims that "both Shagrir and Sprevak are correct when they point out that the logical status of a gate is indeterminate prior to the attribution or identification of its representational content" (2018, 107). They both insist, however, that this individuation scheme is *not computational*.

[15]See, e.g., the discussions in Aaronson (2013, 10.6) and Maroney and Timpson (2018). The label "*objet trouvé* computation" is borrowed from the latter.

[16]For some examples of Turing machines solving non-Turing computable problems, see the discussions in Shapiro (1982), Rescorla (2007), Copeland and Proudfoot (2010), Shapiro (2017) and Quinon (2018). For an example of a symbolic (deviant) interpretation that gives rise to a machine computing very easily a hard primality test, see Stalnaker (1999, 261).

[17]At least, according to the view that the domain of the interpretation comprises directly the physical types and not some intermediate abstract/functional structure.

this mapping (or 'tracking').[18]According to this view, a mapping relation from a physical to a formal structure can be an implementation but *not* an interpretation. The implementation relation would make an interpretation relation when the physical magnitudes have the teleological function to track or stand-for the target's formal entities. These claims are controversial, however. If one thinks that interpretation is no more than a homomorphic, pattern-preserving, relation, or that implementation *itself* requires representation, then the two notions are rather similar, if not identical (for further discussion, see, e.g., Dresner 2010).

To sum up, we characterized and distinguished between two kinds of computational indeterminacy. One kind is a functional indeterminacy, which has to do with different groupings of physical properties into different abstract structures. The other kind of indeterminacy has to do with the interpretation (most would say the formal content) of the physical/abstract states of the system.

We close this section by straightening out a final piece of business. We do not claim that both kinds of indeterminacy arise for every system. First, there may be systems that exhibit functional but not interpretative indeterminacy. For example, it is conceivable that there may exist (say) tri-stable systems that give rise to more than one different functional profile, such that each of them can nonetheless be interpreted by the same, self-dual Boolean function.[19] Due to the self-duality of the interpreted function, then, the system can only be functionally indeterminate. Second, there can be systems that exhibit interpretative but not functional indeterminacy. For example, a logic gate bought from an electronic store or built with vacuum tubes may be used as an AND-gate in a toy circuit (if the existence of a signal in the input/output channels is interpreted as truth-value T) or as an OR-gate (if the existence of a signal in the input/output channels is interpreted as truth-value F). At the same time, it is conceivable that the dynamical behavior of the gate is such that a preferred functional profile can be singled out without the need for external (contextual) restrictions, other than the physics of the system itself. Thus, there can be systems that exhibit both, only one, or neither kind of the computational indeterminacies into examination here.

We turn next to discuss how the two different kinds of indeterminacy bear on the relation between levels and computation.

3 Levels of description and the two kinds of indeterminacy

The relationship between computation and levels of description is a complex one, as it depends on one's view about levels and on one's views about computation (for a review see Elber-

[18]See, e.g., Shea (2018) and Morgan and Piccinini (2018).

[19]A self-dual Boolean function is one that remains the same when interchanging its True and False values. Differently put, a self-dual function is equivalent to its dual one.

Dorozko and Shagrir 2018). On one familiar tri-level explanatory framework, it is best to account for the behavior of a complex computing system in terms of three levels of *description*.[20] According to this framework, the first, higher level is a semantic (Pylyshyn, 1984), knowledge (Newell, 1980, 1982), or intentional (Dennett, 1971) level.[21] This top level has to do with the content (or interpretation) of the system's states. The second, intermediate functional (or symbolic or syntactic, or design[22]) level provides an abstract, more formal, description of the structure of the system, specifically of the relations and the transition dynamics (algorithms) over the states, which encode information. The third, bottom physical (or implementational) level specifies how the entire system is realized in physical or biological structures. Many authors identify the *computational* level with the symbolic level (Pylyshyn 1984, Fodor 1994, Chalmers 2011);[23] others with the semantic and symbolic levels (Block, 1986).[24] Although practically all interlocutors in the debate agree that each level constrains and is constrained by other levels, it is controversial to what extent each level is autonomous.

At first glance, we can easily locate the functional and interpretative indeterminacies within this overall picture of levels. Functional indeterminacy emerges when entities at some physical level[25] are typed together at a higher level of functional organization. This step does not require interpretation in the sense used above. Interpretative indeterminacy emerges when entities at either the physical or the higher functional organization level are assigned some semantic content. Assuming for a moment that the interpretation is assigned to abstract (functional) types, then the two indeterminacies can be seen as composing an inter-level hierarchy: functional indeterminacy emerges in moving from the physical to the functional level, and interpretative indeterminacy emerges further up, in moving from the functional to the semantic level.

[20]We note that the term 'levels of description' does not imply that such levels are not *ontological*. Quite the contrary, most authors take it that these descriptions express real properties of (real) processes and events (the exception is Dennett 1971, who is less committal with respect to the intentional level). Other familiar frameworks of levels are in terms of levels of *organization* (Churchland and Sejnowski, 1992) and levels of *mechanisms* (Craver, 2007), where both approaches understand inter-level relations as part-whole relations.

[21]Most authors cited above have in mind a *cognitive* system.

[22]Newell (1982) terms the second level 'symbolic'. Note that both Newell and Pylyshyn (who uses the term 'syntactic') have in mind "classical" digital systems. Dennett uses the term 'design'.

[23]We can include here the mechanists, who characterize computational properties as medium-independent (e.g., Piccinini 2015 and Coelho Mollo 2018, 2019).

[24]Marr (1982) tri-level picture is an interesting case. He terms the top-level as the *computational* level, and the second level as the *algorithmic* level. Some interpreters of Marr say that his top level specifies in formal terms the input-output function, whereas the algorithmic level specifies, in abstract terms, the mediating mechanism (Egan 1995, Ramsey 2007). They thus think that his computational plus algorithmic levels are similar to the symbolic level. Others think that Marr's top (computational) level includes semantic elements (see Bechtel and Shagrir 2015 for a review of these interpretations).

[25]This does not have to be a physical level in some fundamental sense, such as a physical description stemming from some fundamental theory of physics (e.g., quantum mechanics). It only needs to concern physical entities and descriptions (e.g., input-output functions) at some level of analysis.

But, in fact, the functional and interpretative indeterminacies show that, when put in the context of computational individuation, the relations between levels are more complex. Consider functional indeterminacy first. The question, at this level, is which of the functional profiles of the physical system constitutes its computational profile (i.e., its computational *identity*). Some claim that all of them do (e.g., Chalmers 2011, Lee 2021). Lee (2021), for one, claims that multiple, hierarchical levels (*narrow* functional, *wide* functional, and semantic) may inform different kinds of computational individuation. Thus, the individuation of computation depends on both the nature of the physical system concerned and our epistemic preferences. Others claim that the most *basic* functional profile of the physical system constitutes its computational identity (e.g., Coelho Mollo 2018). Some argue that *contextual* clues have to be resorted to in order to pin down the computational profile. Among this last group, there are those who characterize the contextual clues in further functional terms (e.g., Piccinini 2015, Fresco and Miłkowski 2019). Nevertheless, there are also others who argue that the relevant computational profile is determined by *semantic* content (notably, Sprevak 2010, and Shagrir 2020). Finally, there are also some who insist that just the implementing physical properties suffice for computational individuation —that is, without appealing to the system's functional profile at all (e.g., Dewhurst 2018). We thus observe that on various different views of computational individuation, the distinction between the physical, functional and semantic levels is not clear-cut at all.

We conclude this section by pointing out two open issues regarding interpretative indeterminacy. The first issue concerns the *domain* of the interpretation relation when the interpreted system is a computing system. If the domain consists of entities at the physical (rather than the functional) level, then the two kinds of indeterminacy cannot be seen as composing an inter-level hierarchy, as mentioned above. Even if we grant that the relations between the physical, functional and semantic levels themselves are hierarchical —although, as we have just described, computational individuation tips the scales against an inter-level picture like this—, the two indeterminacies can be seen as being rather orthogonal. This is because each kind of computational indeterminacy involves inter-level relations between the physical and some higher level; and yet the higher level is a *different one* for each of the two kinds of indeterminacy. As a result, if the domain of the interpretation relation pertains directly to entities at the physical level and, even more, the inter-level relations are *not* hierarchical, then a fortiori the two indeterminacies are not hierarchical either.

The second issue is conditional: if there is more than one possible interpretation, which one is relevant to the computational profile of the physical system? Some claim that none of the available interpretations is —e.g., Coelho Mollo 2018, Dewhurst 2018— thereby preserving the distinction between the functional (computational) and semantic levels. Others, however, claim that the determining factors are the same *functional* clues that are relevant to removing functional indeterminacies (Piccinini 2015, Fresco and Miłkowski 2019), and, yet, others argue that the determining factors are the same *semantic* clues that are relevant to removing functional indeterminacies (Sprevak 2010, Shagrir 2020). Lastly, Hemmo and Shenker (2019) argue that

a solution to any indeterminacies is ultimately to be provided by the fundamental physics of the system under consideration: "The view we put forward here, as a solution to the multiple-computations problem ... is in the framework of a reductive physicalist identity theory of mind" (p.104). They thus take it that the picture of levels —in the sense of different ontological levels— is misguided, and offer instead a picture along the lines of "flat physicalism".[26]

Once again, then, quite a few views on computational individuation imply that the relation between the two kinds of indeterminacy and the separation between the physical, functional and semantic levels is not straightforward. That being said, the fact remains that in pointing out that the challenge of computational indeterminacy involves two different (inter-level) dimensions we gain a better understanding of the many contours of individuating a computing system on the basis of different levels of description.

4 Concluding remarks

We distinguished between two kinds of indeterminacy that are directly related to natural computation and computational individuation; a *functional* indeterminacy and an *interpretative* indeterminacy. In pointing out that what has traditionally been regarded as a unified challenge is actually two separate phenomena —each of a different underlying nature— a finer-grained perspective on the challenges posed by the interrelationships between computational individuation, implementation, and levels of organization is gained. Furthermore, we have attempted to indicate that the two kinds of indeterminacy, taken at their limits, can be seen as underlying two important, unrelated thus far, problems in philosophy of computation: triviality arguments and deviant encodings. The former can be seen as a limiting case of functional indeterminacy, whereas the latter can be seen as a limiting case of interpretative indeterminacy.[27] Finally, we have examined how the different kinds of indeterminacy relate to different description levels of computational systems. The upshot of our discussion is that the fine-grained structure of this relation does not admit of some straightforward account, insofar as it is unmediated by a precise account of computational individuation. Thus, the road to pinning down how the different indeterminacies bear on the inter-level relations within computational systems passes through the straits of developing a satisfactory theory of computational individuation.

[26] See Shenker (2017).

[27] To take these kinds of indeterminacy at their limits means here to consider them without any additional constrains on the extent to which they underpin computational individuation; that is, to consider functional indeterminacy unconstrained by any modal, counterfactual, causal (or other) parameters, and to consider interpretative indeterminacy unconstrained by any semantic, computational, representational (or other) parameters.

Acknowledgments

We are thankful to two anonymous referees for their comments and suggestions. This research was supported by the Israel Science Foundation Grant (830/18).

References

Aaronson, S. (2013). Why philosophers should care about computational complexity. In B. J. Copeland, C. J. Posy, and O. Shagrir (Eds.), *Computability: Turing, Gödel, Church, and Beyond*, pp. 261–328. The MIT Press.

Bechtel, W. and O. Shagrir (2015). The non-redundant contributions of marr's three levels of analysis for explaining information-processing mechanisms. *Topics in Cognitive Science 7*(2), 312–322.

Block, N. (1986). Advertisement for a semantics for psychology. *Midwest Studies in Philosophy 10*(1), 615–678.

Burge, T. (1986). Individualism and psychology. *The Philosophical Review 95*(1), 3–45.

Chalmers, D. (1996). Does a rock implement every finite-state automaton? *Synthese 108*(3), 309–333.

Chalmers, D. (2011). A computational foundation for the study of cognition. *Journal of Cognitive Science 12*(4), 323–357.

Chalmers, D. (2012). The varieties of computation: A reply. *Journal of Cognitive Science 13*(3), 211–248.

Churchland, P. and T. Sejnowski (1992). *The Computational Brain*. Cambridge, Massachusetts.

Coelho Mollo, D. (2018). Functional individuation, mechanistic implementation: The proper way of seeing the mechanistic view of concrete computation. *Synthese 195*(8), 3477–3497.

Coelho Mollo, D. (2019). Are there teleological functions to compute? *Philosophy of Science 86*(3), 431–452.

Copeland, B. J. and D. Proudfoot (2010). Deviant encodings and turing's analysis of computability. *Studies in History and Philosophy of Science Part A 41*(3), 247–252. Computation and cognitive science.

Craver, C. F. (2007). *Explaining the Brain: Mechanisms and the Mosaic Unity of Neuroscience*. Oxford University Press, Clarendon Press.

Dennett, D. C. (1971). Intentional systems. *Journal of Philosophy 68*(February), 87–106.

Dewhurst, J. (2018). Individuation without representation. *The British Journal for the Philosophy of Science 69*(1), 103–116.

Dresner, E. (2010). Measurement-theoretic representation and computation-theoretic realization. *The Journal of Philosophy 107*(6), 275–292.

Dretske, F. I. (1988). *Explaining Behavior: Reasons in a World of Causes*. MIT Press.

Duwell, A. (2018). How to make orthogonal positions parallel: Revisiting the quantum parallelism thesis. In M. Cuffaro and S. Fletcher (Eds.), *Physical Perspectives on Computation, Computational Perspectives on Physics*, pp. 83–102. Cambridge University Press.

Egan, F. (1995). Computation and content. *The Philosophical Review 104*(2), 181–203.

Egan, F. (2012). Metaphysics and computational cognitive science: Let's not let the tail wag the dog. *Journal of Cognitive Science 13*(1), 39–49.

Elber-Dorozko, L. and O. Shagrir (2018). Computation and levels in the cognitive and neural sciences. In M. Colombo and M. Sprevak (Eds.), *Routledge Handbook of the Computational Mind*, pp. 205–222. Routledege.

Fodor, J. A. (1994). *The Elm and the Expert: Mentalese and Its Semantics*. MIT Press.

Fresco, N. (2015). Objective computation versus subjective computation. *Erkenntnis 80*(5), 1031–1053.

Fresco, N., B. J. Copeland, and M. J. Wolf (2021). The indeterminacy of computation. *Synthese*. https://doi.org/10.1007/s11229-021-03352-9.

Fresco, N. and M. Miłkowski (2019). Mechanistic computational individuation without biting the bullet. *The British Journal for the Philosophy of Science*. https://doi.org/10.1093/bjps/axz005.

Haugeland, J. (1978). The nature and plausibility of cognitivism. *Behavioral and Brain Sciences 1*(2), 215–226.

Hemmo, M. and O. Shenker (2019). The physics of implementing logic: Landauer's principle and the multiple-computations theorem. *Studies in History and Philosophy of Science Part B: Studies in History and Philosophy of Modern Physics 68*, 90–105.

Lee, J. (2021). Mechanisms, wide functions, and content: Towards a computational pluralism. *The British Journal for the Philosophy of Science 72*(1), 221–244.

Maley, C. J. and G. Piccinini (2017). A unified mechanistic account of teleological functions for psychology and neuroscience. In D. M. Kaplan (Ed.), *Explanation and Integration in Mind and Brain Science*, pp. 236–256. Oxford University Press.

Maroney, O. J. and C. G. Timpson (2018). How is there a physics of information? On characterizing physical evolution as information processing. In M. Cuffaro and S. Fletcher (Eds.), *Physical Perspectives on Computation, Computational Perspectives on Physics*, pp. 103–126. Cambridge University Press.

Marr, D. (1982). *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W.H. Freeman and Company, NY.

Matthews, R. J. and E. Dresner (2017). Measurement and computational skepticism. *Noûs 51*(4), 832–854.

Millikan, R. G. (1984). *Language, Thought, and Other Biological Categories: New Foundations for Realism*. MIT Press.

Morgan, A. and G. Piccinini (2018). Towards a cognitive neuroscience of intentionality. *Minds and Machines 28*(1), 119–139.

Newell, A. (1980). Physical symbol systems. *Cognitive science 4*(2), 135–183.

Newell, A. (1982). The knowledge level. *Artificial intelligence 18*(1), 87–127.

Piccinini, G. (2015). *Physical Computation: A Mechanistic Account*. Oxford University Press, USA.

Putnam, H. (1967). Psychological predicates. In W. H. Capitan and D. D. Merrill (Eds.), *Art, Mind, and Religion*, pp. 37–48. University of Pittsburgh Press. Reprinted as The nature of mental states in his *Mind, Language and Reality, Philosophical Papers*, vol. 2, Cambridge University Press, pp. 429–440.

Putnam, H. (1988). *Representation and Reality*. MIT press.

Pylyshyn, Z. W. (1984). *Computation and Cognition: Toward a Foundation for Cognitive Science*. MIT Press.

Quinon, P. (2018). A taxonomy of deviant encodings. In F. Manea, R. G. Miller, and D. Nowotka (Eds.), *Sailing Routes in the World of Computation*, pp. 338–348. Cham: Springer International Publishing.

Ramsey, W. M. (2007). *Representation Reconsidered*. Cambridge University Press.

Rescorla, M. (2007). Church's thesis and the conceptual analysis of computability. *Notre Dame Journal of Formal Logic 48*(2), 253 – 280.

Rescorla, M. (2014). A theory of computational implementation. *Synthese 191*, 1277–1307.

Searle, J. R. (1992). *The Rediscovery of the Mind*. MIT press.

Shagrir, O. (2001). Content, Computation and Externalism. *Mind 110*(438), 369–400.

Shagrir, O. (2020). In defense of the semantic view of computation. *Synthese 197*, 4083–4108.

Shapiro, S. (1982). Acceptable notation. *Notre Dame Journal of Formal Logic 23*(1), 14 – 20.

Shapiro, S. (2017). Computing with numbers and other non-syntactic things: De re knowledge of abstract objects. *Philosophia Mathematica 25*(2), 268–281.

Shea, N. (2018). *Representation in Cognitive Science*. Oxford University Press.

Shenker, O. (2017). Flat physicalism: Some implications. *Iyyun: The Jerusalem Philosophical Quarterly 66*, 211–225.

Sprevak, M. (2010). Computation, individuation, and the received view on representation. *Studies in History and Philosophy of Science Part A 41*(3), 260 – 270.

Stalnaker, R. C. (1999). *Context and Content: Essays on Intentionality in Speech and Thought*. Oxford University Press.

Turner, R. (2021). Computational abstraction. *Entropy 23*(2), 213. https://doi.org/10.3390/e23020213.