Bulletin of Taras Shevchenko National University of Kyiv.
Philosophy, 1(8), 5-8.
UDC 16/164:007+004
https://doi.org/10.17721/2523-4064.2023/8-1/8

Mohamad Awwad, Dr. of Philosophy, Independent Researcher
https://orcid.org/0000-0002-0699-5454
e-mail: awwad.mohamad@gmail.com

ON COMPUTATIONALISM: FORMAL INTERPRETATION AND INITIAL MODEL

In this article, we propose an initial formal model of computationalism based on mathematical relations between cognition and computation. More specifically, based on a set of cognitive constituents as a domain, and a set of computational implementations as a range, we define two relations of transformation over these sets. Moreover, we define the principles of implementability, describability, and phenomena correspondence, and we conjecture that full computationalism does not hold since these principles are not fulfilled. Particularly, many cognitively-tied phenomena fail to respect the describability principle which is necessary for representing a cognitive state by a computational state.

Keywords: Full/Partial computationalism; cognition; principle of implementability; principle of describability; formal computationalism; implementation transformation; phenomena transformation.

Introduction. Computationalism claims that cognitive states are computational states. This view is one among many others in the computational theory of mind. Computational processes are said to 1) explain 2) model, or 3) capture cognitive phenomena. In this work, we interpret full computationalism as if we are able to provide computable implementations to cognitive constituents.

Referring to computable implementations necessitates specifying the computability theory we are invoking, since any computational interpretation of cognition is bounded by its model of computation.

For instance, as Turing machines model does not integrate non Turing-computable functions, it would be legitimate to wonder if this model is the most adequate to adopt in this discussion, or, oppositely, there is another model more fundamental to consider. As another example, models of hypercomputation (Bringsjord & Zenzen, 2012; Franchette, 2011; Syropoulos, 2008) such as analog recurrent neural network and infinite-time Turing machine have been proposed as model of computation. Consequently, hypercomputationalism would become the position claiming that hypercomputation can capture or model cognition.

Some scholars go even further in proposing pancomputationalism that considers almost everything, including human mind, as computational. This claim is, for us, too general and ambiguous; however we here prefer uniquely focus on the relationship between cognition and computation.

Now among all these models and interpretations, computationalism could be valid or not depending on the version of computationalism we propose (e. g. weak, strong, etc.). For example, the assumption that mental states are equivalent to computational states implies a strong version of computationalism. However, it could always be possible to reject the strong computationalism view (Searle, 1984, 2008) and defend a weaker version of it that claims computation can *partially* implement cognition or some cognitive phenomena in terms of these phenomena's "byproduct".

We should also clarify the kind of activities or phenomena that have to be included in the notion of cognition in order to investigate their potentiality in being computational activities. For instance, as we generally consider perception as a cognitive process, we think we should take into consideration the likely cognitive scope that emotions can have (LeDoux & Brown, 2017). Therefore the full computationalism position needs evidence on how emotional feelings could be computational states. In addition, we can also wonder the exact place that instinct and unconscious intuitive components could take in the cognitive system. Suppose we consider these concepts cognitive or cognitively-tied concepts that are part of cognition, act directly or indirectly, and consciously or unconsciously on cognition. To what extent can we grasp the constituents of these notions in order to satisfactorily capture them in a computational manner? In this paper, we effectively consider emotions or emotional feelings as cognitively-tied states and directly involved within the cognitive activities. We use this view, among others, in order to highlight the immense difficulties to capture all and every cognitive constituent by computational processes.

More importantly, full computationalism should require that the constituents of cognition and their phenomena are represented, and interpreted by computational manner. It is not satisfactory to claim that cognition is computational because some of its components resemble to an input-output processing. Indeed, we should require showing how every element, activity, and phenomenon of cognition corresponds to a computational element, activity, and phenomenon, as well as captured by a specific theory of computation. Any model of computationalism must be precisely defined in order to start searching for a potential correspondence between cognition and computation.

Consequently, in this article, we propose that full computationalism requires *computational correspondences* with every single cognitive constituent and phenomenon. For instance, to prove that a specific cognitive activity that deals with learning is computational, we should at least show that the cognitive functions and outcome of this learning activity as well as the involved cognitive phenomenon in the learning activity have their computational corresponding counterparts. We refer to these kinds of requirements when we try to interpret formally the notion of computationalism in section 2.

We are, nevertheless, aware of the immense difficulties of precisely and exhaustively capturing cognitive constituents and phenomena. These difficulties could, at first glance, appear almost impossible to overcome. Indeed, we think they are. Arguably, these difficulties, in turn, point to the corresponding challenges faced by the proponents of the full computationalism position.

These stringent definitional requirements characterize the full or strong view of computationalism as we will be referring to in the rest of the article. Similarly, partial or weak computationalism requires only a part of cognitive activities to be implemented by computational processes. The word *computationalism*, used alone, refers to the position of full computationalism.

Methodology. This subject is purely theoretical, and our methodology in this work is analytic-synthetic.

Results.

1. The general settings of the model

In this section, we propose an initial model of computationalism. It aims at interpreting to some extent this notion in a formal manner. As a model settings requirement, when claiming any position about computationalism, it could be beneficial to specify a frame within which this notion could be investigated. These settings can include the following precisions:

1) The meaning of computationalism which has the following form:

Computation RELATES Cognition, where:

$$\text{RELATES} = [\text{IsEquivalent}|\ \text{Models}|\text{Explains}|\text{Implements}|\ldots].$$

2) The computability model (i. e. Turing machine, biological, quantum, hypercomputational, etc.) referred to when using "computationalism". This should directly imply the meaning of computational processes and an interpretation of the computable cognitive processes.

3) The extent to which we can provide an interpretation of the constituents or the components of cognitive phenomena. It seeks identifying the elements involved in a cognitive phenomenon at the lowest possible level of interactions.

This interpretation and identification process mentioned in point 3 has probably the most significant consequence on the problem of computationalism. It directly unveils whether these cognitive components are parts of computable cognitive phenomena or not. These precisions are the most difficult to determine and the elements of this set are particularly difficult to define in a precise manner. In our opinion, the reason is due in part to the elusive appearance of some elements' types that belong to indescribable cognitive phenomena; they are constituted of so many complex, ambiguous, and ungraspable elements which escape satisfactory precision.

2. A Model of Computationalism

In this section we propose the following *7-tuple* of computationalism that need to be identified.

$$Computationalism(S, COGNITION, DESC, COMP, PHEN, T, R)$$

a) Settings

The settings S contains the three-point model settings we just discussed; we make the following assumptions in our proposed model:

1) We understand by computationalism that *computation implements cognition* with respects to the requirements we specified in the introduction.

2) We consider the Turing machine model as the computability model.

3) We intuitively refer to a cognitive component as any element involved at any physical (biochemical, electromagnetic, etc.) or abstract level of interactions that produce a mental phenomenon.

b) The four sets: COGNITION, DESC, COMP, and PHEN

We consider COGNITION the set of any single element of cognition. That is to say any elementary cognitive component (set in the preceding point 3) that acts within the cognitive phenomena.

In this context, we can define $DESC$ (for description) as the set of all, possibly infinite, elementary cognitive descriptions (or describable components of cognition), and $COMP$ (for computation) a set of elementary computable implementations. The set $PHEN \subset 2^{COGNITION}$ (representing the cognitive phenomena) includes any subset of COGNITION that is possible to *happen* within the cognition as either consciously or unconsciously experienced phenomenon.

Remark on the implementable, describable, and indescribable phenomena. As mentioned before, it is difficult to define an elementary cognitive description without a certain ambiguity. An elementary description could be defined as *a unitary statement that emanates from the experience of a cognitive phenomenon*. We used the word "experience" to include not only what the statement itself expresses but also what the phenomenon can reach when applying or "experiencing" what the statement says. For example, "I know what it is like to be a human", "I remember when I was 13-year old", and "I saw a blue light" could be interpreted as elementary descriptions. These three statements are indeed considered elementary descriptions written in unitary statements, but they inexpressibly designate different phenomena that are cognitively experienced but not necessarily encapsulated by these unitary statements. For example, what are the invoked cognitive experiences in knowing *what it is like to be a human*? What are and how I am collecting my memories when I was 13-year old? And finally, how did I really see a blue light? And what are the steps of interpretation that my cognition did (meaning I did) to conclude what I saw, and the repercussions I experienced in seeing it?

The challenge of identifying the experienced elements of a cognitive phenomenon can be also applied to emotional feelings. The elements of these cognitively-tied feelings seem to escape precise descriptions. Consequently, no computational implementation would be possible in capturing these cognitive elements.

Briefly, we try to point out all these cognitively-tied experiences that are not only expressed by informative written statements, but need detailed descriptions that seem going beyond any natural language or beyond any mental capacity. This view suggests three types of cognitive phenomena:

1) The describable and implementable.

2) The describable by a unitary statement but not implementable because there are no enough detailed descriptions of the phenomenon (the 3 examples given previously).

3) The indescribable (some components of the phenomenon that do not have descriptions by unitary statements; note that this last statement describes to which set these phenomena belong without describing anything about them!).

Consequently, some describable components (belong to DESC) are implementable but not necessarily all of them. The relation between a description and an implementation will be formally clarified in the next lines.

Now, we complete the *7-tuple* announced model of computationalism by defining the relations $T\ and\ R$.

c) The relation T of implementation transformation

Let $x \in\ COGNITION\ and\ \ y \in\ COMP, xTy\ \subseteq COGNITION \times COMP$

$if\ and\ only\ if\ "y\ is\ an\ implementation\ of x".$

d) The relation R of phenomena transformation

$$if\ p \in 2^{DESC}, q \in 2^{COMP}, the\ relation\ R \subseteq 2^{DESC} \times 2^{COMP} is\ defined\ as\ pRq\ if\ and\ only\ if\ "the\ process\ q\ captures\ the\ cognitive\ phenomenon\ p".$$

An illustrative example of the relations T and R over the discussed sets is given in Figure 1.

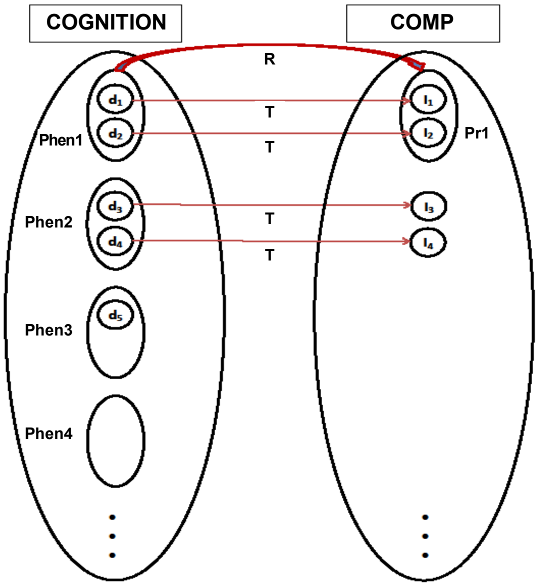

Figure 1. Example of the relations T and R

Note that Phen1 has two descriptions d_1 and d_2 that are implementable by I_1 and I_2 (represented by the relation T), and has a phenomenon correspondence (represented by the relation R). Phen2 has also two descriptions d_3 and d_4 that are implementable by I_3 and I_4, but it does not have a phenomenon correspondence. Phen3 has one description d_5 which is not implementable, and finally Phen5 has no descriptions at all which means it belongs to the indescribable phenomena.

3. On computationalism using the formal model

As the main elements of the model are set, we can interrogate this model on the validity of computationalism. We need to define for this purpose the following principles:

The principle of implementability

A cognitive description ($\in DESC$) is implementable if there is a relation of implementation that transforms it into an elementary computable implementation. This means:
$(\exists x \in DESC, \exists y \in COMP, xTy) \rightarrow (x\ is\ implementable),$
$where\ T\ is\ the\ relation\ of\ implementation\ transformation.$

The principle of describability

A cognitive phenomenon ($\subset PHEN$) is describable if it can be decomposed into a set of elementary cognitive descriptions ($\subset DESC$).

The principle of phenomena correspondence

A cognitive phenomenon has a corresponding computable process if it has a relation of phenomena transformation; that is to say,

$$(\exists p \in PHEN, \exists q \subset 2^{COMP}, pRq) \rightarrow (p\ has\ a\ computable\ process)$$

$where\ R\ is\ the\ relation\ of\ phenomena\ transformation.$

To illustrate the importance of phenomena correspondence, we can consider the analogy of replicating a physical object at a certain level of precision. This replication should take into consideration not only a correspondence of their elementary constituents (at a molecular level, for example), but also should guarantee a holistic correspondence between the original object and its replication in terms of their functions, objectives, etc. Similarly, for a cognitive phenomenon, we should also require a potential computational correspondence as a whole (functioning, expectations, etc.) in addition to a correspondence of its elementary instructions.

Definition 1: The full computationalism position consists of admitting the following statements:

1) The principle of describability holds for all cognitive phenomena in PHEN.

2) The principle of implementability holds for all cognitive descriptions in DESC.

3) The principle of phenomena correspondence holds for all cognitive phenomena in PHEN.

Definition 2: The partial computationalism position consists of admitting the following statements:

1) $\exists x \in DESC, \exists y \in COMP, xTy\ where\ T\ is\ the\ relation\ of\ implementation\ transformation.$

2) $\exists p \in 2^{DESC}, \exists q \subset 2^{COMP}, pRq\ where\ R\ is\ the\ relation\ of\ phenomena\ transformation.$

Proposition 1: The partial computationalism position holds.

Obviously, this proposition is true based on the preceding definitions. Logic-based processes constitute, among others, examples of cognitive activities that satisfy the previous definition. In fact, logical calculi, deductions, formal calculations, and knowledge representation are cognitive activities that their elements can be computationally implemented. In addition, their cognitive phenomena have correspondences into computable processes.

Conjecture 1: Some cognitive descriptions in DESC do not satisfy the principle of implementability

Remark: a description could represent only the appearance of the phenomena, but not necessarily its complete elements that allow an implementation. For example, the sentence "I feel happy" is a declaration of feeling, but it is neither a description of exactly *what I feel* nor how the many underpinned cognitively-related interactions produced to me the feeling of happiness.

Conjecture 2: Some cognitive phenomena in PHEN do not satisfy the principle of describability

Remark: we think that the indescribable cognitive phenomena constitute, probably by far, the biggest set of cognition's elements. It certainly includes all these internal hidden phenomena that produced by complex interactions, and that *we live* but we are sometimes even not conscious about their existence.

Conjecture 3: Some cognitive phenomena in PHEN do not satisfy the principle of phenomena correspondence.

Remark: this conjecture is straightforward using the two preceding conjectures. If some descriptions are not implementable, or some phenomena are indescribable, it follows that for some phenomena there are no correspondences into implementable processes.

Conjecture 4: The full computationalism position does not hold.

This conjecture is a direct implication from any of the three preceding conjectures.

Discussion and Conclusion. In this article, the main objective was to propose an initial formal model of computationalism. We submitted a model that emphasizes the mathematical relations between the elements of cognition and those of computation as well as between cognitive phenomena and computational processes. These relations should be satisfied for all elements of their domain in order to prove the validity of full computationalism. We substantiated some non-negligible difficulties such a notion can face insofar as many requirements, such as the principle of describability, are not fulfilled.

Many questions in this work remain open or need more precisions. For example, it will be important to define more precisely the sets COGNITION, DESC, and PHEN. Similarly, we need to know the exact criteria that make a cognitive component describable, and sometimes implementable. If some components are describable and implementable, and some others are not, can we talk about a degree of computationalism? As we assume that many cognitive processes can be seen as computational processes under a specific interpretation, we can interrogate the extent to which partial computationalism is valid. That is to say, we could try to estimate a *degree* of computationalism that represents the scope of cognitive processes that are computationally implementable, i.e., formally, to find a degree $d(COGNITION) \in [0,1]$ of real numbers, where $d = 1$ corresponds to full computationalism, and $d = 0$ means nothing cognitive is computational.

References

Blackmon, J. C. (2007). Computationalism and the Putnam-Searle Thesis. University of California, Davis.

Brey, P., & Søraker, J. H. (2009). Philosophy of computing and information technology. In Philosophy of technology and engineering sciences (pp. 1341–1407). Elsevier.

Bringsjord, S., & Zenzen, M. (2012). Superminds: People Harness Hypercomputation, and More. Springer Science & Business Media.

Davis, M. (2004). The myth of hypercomputation. In Alan Turing: Life and legacy of a great thinker (pp. 195–211). Springer.

Dennett, D. C. (2014). Intuition Pumps And Other Tools for Thinking. W. W. Norton & Company.

Denning, P. J., & Tedre, M. (2019). Computational Thinking. MIT Press.

Dreyfus, H. L., Dreyfus, S. E., & Athanasiou, T. (1986). Mind Over Machine: The Power of Human Intuition and Expertise in the Era of the Computer. NY Free Press.

Floridi, L. (2002). Philosophy and Computing: An Introduction. Routledge.

Floridi, L. (2013). The Philosophy of Information. OUP Oxford.

Fodor, J. (1975). The Language of Thought. Cambridge, MA: Harvard. University Press.

Franchette, F. (2011). Why to Build a Physical Model of Hypercomputation. N. A. Publishing.

Fresco, N. (2012). The explanatory role of computation in cognitive science. Minds and Machines, 22, 353–380.

Horgan, T., & Tienson, J. (2012). Connectionism and the Philosophy of Mind. Springer Science & Business Media.

LeDoux, J. E., & Brown, R. (2017). A higher-order theory of emotional consciousness. Proceedings of the National Academy of Sciences, 114(10). https://doi.org/10.1073/pnas.1619316114

Miłkowski, M. (2018). Objections to Computationalism: A Survey. Roczniki Filozoficzne, 66(3), 57–75. https://www.ceeol.com/search/article-detail?id=700677

Piccinini, G. (2009). Computationalism in the Philosophy of Mind. Philosophy Compass, 4(3), 515–532.

Piccinini, G. (2015). Physical computation: A mechanistic account. OUP Oxford.

Powers, T. M. (2017). Philosophy and Computing: Essays in Epistemology, Philosophy of Mind, Logic, and Ethics. Springer.

Pylyshyn, Z. W. (1984). Computation and cognition. MIT press Cambridge, MA.

Scheutz, M. (2002). Computationalism: New Directions. MIT Press.

Searle, J. R. (1984). Minds, Brains and Science: The 1984 Reith Lectures. British Broadcasting Corporation.

Searle, J. R. (2008). Philosophy in a New Century: Selected Essays. Cambridge University Press.

Siegelmann, H. T. (2012). Neural Networks and Analog Computation: Beyond the Turing Limit. Springer Science & Business Media.

Syropoulos, A. (2008). Hypercomputation: Computing Beyond the Church-Turing Barrier. Springer Science & Business Media.

Turing, A. M. (1950). Computing machinery and intelligence. Mind, 59(October), 433–460.

William, D. (1984). On Some Fundamental Distinctions of Computationalism. University of Western Ontario, Centre for Cognitive Science.

Отримано: 12.01.2023
Ухвалено до друку: 27.03.2023
Опубліковано: 29.03.2023

Вісник Київського національного університету імені Тараса Шевченка. Філософія, 1(8), 5-8.
УДК 16/164:007+004
https:// doi.org/10.17721/2523-4064.2023/8-1/8

Мохамад Аввад, д-р філософії, незалежний дослідник
https://orcid.org/0000-0002-0699-5454
e-mail: awwad.mohamad@gmail.com

ПРО КОМП'ЮТАЦІОНАЛІЗМ: ФОРМАЛЬНА ІНТЕРПРЕТАЦІЯ ТА ПОЧАТКОВА МОДЕЛЬ

Запропоновано початкову формальну модель комп'ютаціоналізму, засновану на математичних зв'язках між пізнанням і обчисленням. Точніше, на основі набору когнітивних складників як області та набору обчислювальних реалізацій як діапазону визначено два види відносин перетворення над цими наборами. Крім того, визначено принципи реалізованості, описуваності та відповідності явищ і зроблено припущення, що повний комп'ютаціоналізм не виконується, оскільки ці принципи не виконуються. Зокрема, багато когнітивно пов'язаних явищ не враховують принцип описання, який необхідний для репрезентації когнітивного стану обчислювальним станом.

Ключові слова: ***повний/частковий комп'ютаціоналізм; пізнання; принцип реалізованості; принцип описовості; формальний комп'ютаціоналізм; трансформація реалізації; трансформація явищ.***