DOI: 10.36446/af.2020.338

THE PROBLEM OF MERGE

El problema de ensamble

Mariela Destéfano [a,b]
https://orcid.org/0000-0003-3145-0306
mariela.destefano@gmail.com

[a] Universidad de Buenos Aires, Argentina
[b] Consejo Nacional de Investigaciones Científicas y Técnicas, Argentina

Abstract

Focusing on biolinguistic considerations, Merge is a formal operation proposed by theoretical linguistics and linked to specific principles of neural computation. In this sense, Merge can be viewed as a natural operation of the brain. Merge is commonly claimed to be a digital operation. In a first approximation, digital computation is the processing of strings of digits according to general rules defined over these digits. However, it seems that neural processes are not digital computations. These conflictive claims, e.g. the digital characterization of Merge and the non-digital characterization of the brain, leads to the following scenario: either Merge is an operation that is not realized in the brain or Merge is realized in the brain but not digitally. The purpose of this paper is to evaluate the problems posed by each thesis.

Key words: Language Faculty; Digital Operation; Analog Operation; Neural Computation

Resumen

Desde la bioloinguistica, ensamble sería una operación digital realizada en el cerebro que, en tanto tal, estaría asociada a principios específicos de la computación neural. En una primera aproximación, la computación digital consiste en el procesamiento de cadenas de dígitos de acuerdo a reglas generales. Sin embargo, los procesos neurales no se desarrollarían de acuerdo a los principios de la computación digital. Estas afirmaciones en conflicto, e.g., la caracterización digital de ensamble y la caracterización no digital del cerebro, llevan al siguiente escenario: o bien ensamble es una operación que no realiza el cerebro, o bien es realizada por el cerebro pero no digitalmente. El propósito de este artículo es evaluar los problemas de estas dos tesis.

Palabras clave

Facultad del lenguaje; Operación digital; Operación analógica; Computación neural.

1. Introduction

Within the biolinguistic approach, theoretical linguistics is biology "at a suitable level of abstraction" (Boeckx & Piattelli-Palmarini, 2005). It studies the Faculty of Language (FL), which is understood to be a "cognitive organ" of the individual mind/brain which shares many of the properties of visual perception, motor action, etc. I-language ("I" signifying individual, internal and intentional) is the computational system of the FL that generates the expressions (logical and phonological forms) that interface with conceptual-intentional and sensory-motor systems. The computational operation of I-language that generates such expressions is called "Merge".

Merge is an operation that combines two syntactic objects α and β to form a complex syntactic object K. Chomsky maintains that the value of K must at least include a label indicating the type to which K belongs (Chomsky, 1995). Logical considerations lead him to conclude that the label of K must be constructed from the two constituents α and β. Accordingly, the value of Merge (α, β) is K, which is either {α,{α, β}} or{β,{α, β}}. Merge allows grammars, being themselves finite, to generate an infinite number of linguistic expressions (Al-Mutairi, 2014). In other words, Merge is the recursive mechanism responsible for the apparent discrete infinity of natural languages in the sense that, when language are thought of as sets of expressions, these sets are infinite. The infinitude claim also involves the idea that there is no limit on the potential length of linguistic expressions ('there is no longest sentence') (Hulst, 2010). Given that many notions of Merge proliferate in the literature, I will restrict my presentation to Chomsky´s characterization.

Focusing on biological considerations, the formal operations proposed at the level of theoretical linguistics are linked to specific principles of neural computation. In this sense, Merge can be viewed as a natural operation of the brain (Boeckx, 2013b). Merge is commonly claimed to be a digital operation. In Chomsky's words, "…we can think of recursion [Merge] as enumeration of a set of discrete objects by a computable finitary procedure, one that can be programmed for an ordinary digital computer that has access to unlimited memory and time" (Chomsky, 2014, p. 1). In a first approximation, digital computation is the processing of strings of digits according to general rules defined over these digits (Piccinini & Scarantino, 2010). This notion of "computation" was inherited from the pioneering works of Turing on computable functions (Turing, 1936). However, it seems that neural computations do not manipulate digital procedures (Piccinini & Bahar, 2013). These conflictive claims, e.g. the digital nature of Merge and the non-digital characterization of the brain, lead to the following scenario: either Merge is best understood as an operation that is not realized in the brain or Merge is realized in the brain but not digitally. This paper aims to evaluate the problems of each thesis and, in particular, it will examine the scope and difficulties of the second thesis in some detail.

The article is organized as follows: Section 2 introduces the digital characterization of Merge, depicting a notion of "digit" and "digital operation" as inspired by Haugeland's work. Section 3 presents my arguments as to why I believe that the brain does not operate in a digital way. For that purpose, I explore certain empirical data related to the electrochemical activity of the brain. Section 4 addresses the thesis that Merge is not realized in the brain and considers a range of methodological issues pertaining to the biolinguistic approach. Section 5 analyzes the thesis that Merge is realized in the brain but not digitally. I suggest that neuronal activity has some analog properties and I discuss the difficulties for an analog system to realize a digital operation. And finally, Section 6 contains the conclusions of the article.

A disclaimer from the start: I am not a linguist, and, aside from a few linguistic issues that I will simple present in the first sections, I do not expect to say anything that turns on details of linguistic theory. The point of my discussion does not need any substantive linguistic commitment.

2. The Digital Characterization of Merge

Merge is taken to be the basic combinatorial operation of I-language (Fukui, 2011). In its most elementary form, Merge is a simple grouping procedure that puts α and β together. It is a *binary operation* that takes two constituents as inputs and combines them to form a novel constituent. Merge operates as a set formation rule, taking two already formed syntactic objects and constructing from them a new object: Merge (α), (β) = (α, β). This set formation operation is recursive, since it can be applied to its own outputs without limit, and it does not require any linear order in the members of the set (Boeckx, 2009). Presumably, this set formation operation is very common across cognitive modules beyond the FL (Fujita, 2017).

Furthermore, Merge is a procedure that gives rise to endocentric structures, since the new already formed object is labeled by one of the inputs. Merge selects one of the two members to be the head of the new construction and the head would be the unit over which Merge operates further combinations. For instance, when you put together a verb and a noun, typically what you get is a verb, and this complex construction acts as a verb in the next combinations. These headed constructions constitute the kind of hierarchical structures that we usually get in language. For instance, Merge composes the following hierarchical structure:

{Mary, loves, books}

First, composing a set that has “loves” and “books” as units and the verb as the head:

{loves {loves, books}}

And, second, forming another set that adds the unit “Mary” and keeps the verb as the head:

{loves {Mary {loves{loves, books}

In biolinguistic literature, Merge is generally considered a *digital operation*. Chomsky claims that “…we can think of recursion [Merge] as enumeration of a set of discrete objects by a computable finitary procedure, one that can be programmed for an ordinary digital computer that has access to unlimited memory and time” (Chomsky, 2014, p. 1) . It should be remarked that Merge yields a potentially infinite set of linguistic structures operating in the way drawn by the successor function (Chomsky, 2008; Kleene, 1952; Lobina, 2014). The successor function is a mathematical engine that underlies the “iterative conception of set”, which means a process in which sets are “recursively generated at each stage” (Boolos, 1971, p. 223). Peano´s definition of natural numbers is the classical *locus*, where the successor function is applied in order to construct a set and its members (Lobina, 2017). Following Kleene (1952) the successor function in natural numbers can be characterized in three steps: (i) 0 is a natural number (this is the basic clause), (ii) if *n* is a natural number, then *n*+1 (or n’) is also a natural number, (iii) all numbers are defined by steps (i) and (ii).

Given that Merge would be mathematically characterized as a successor function and successor function is a computable function, its best explanation would to understand it as a digital operation. Systems which transform structures such as $\vec{\varnothing}_{1,}$ **...** $\vec{\varnothing}\mathbf{n}$ into structures such as $\vec{\varnothing}\mathbf{n}_{\ +1}$ are best understood to manipulate rules that consider such expressions as strings of digits. So, what then, would a digital operation be? The most straightforward notion is that of a "digital operation" in reference to digital computation. In referring to digital computation, Turing´s notion of computability is commonly mentioned, since Turing computation is often described as digital. It is commonly conceded that Turing computation captures the manner in which every mechanical device computes a calculable function such as successor function (Lobina, 2011). He proposed a theoretical notion of computability in order to answer the *Entscheidungsproblem*, or the question of whether, given a formal system, there is a general method for deciding if a well-formed formula is a theorem of a system. Church and Turing both answered this question in the negative (King, 1996).

In this context, digital computation consists in the processing of strings of digits according to rules defined over the digits (Piccinini & Scarantino, 2010). This kind of processing is usually seen as performed by a Turing machine. A Turing machine is an abstract structure with which Turing reduced a computational process to its essentials (De Mol, 2018). The machine has, in part, an infinite tape divided into cells, each of which contains a digit (either "0" or "1") and a read-write head which scans a single cell on the tape. The action of a Turing machine is determined by the current state of the machine, the digit in the cell currently being scanned and a table of transition rules (De Mol, 2018). Transition rules can be understood as procedures that state the following: "if the machine is in current state *X* and the cell being scanned contains the digit *Y*, then move to the next state *Z* taking action". Actions are the outputs of the transition rules and consist either in writing a new digit ahead on the tape or moving the head to the right or left, selecting an already written digit.

With these transition rules in mind, a digital operation is minimally characterized as a computational rule that maps inputs to outputs manipulating strings of digits. This sort of rule for digital computation is simply a map from the input string of digits to the output string of digits, plus some internal states. Examples of operations that may figure in a digital computation include addition, multiplication, sorting, etc. In the case of a Turing Machine, transition rules are explicit and algorithmic, in the sense that the machine has a finite list or table of executable instructions which specify a certain order of steps to achieve a result (Copeland, 1996). However, I am presenting a minimal notion of a digital operation, which does not require that the rules defining the computation be represented in the computing system, nor that the rules constitute a fixed algorithm followed step-by-step by a computing system. All that the rules need to do is specify what relationship is obtained between the strings of digits (Piccinini, 2007). This minimal notion of a digital operation is enough to characterize Merge, which is a procedure that needn't perform the transformations in any sequential order (Boeckx, 2009).

What kind of entities are the digits manipulated in digital computation? This question arises in order to specify the nature of a digital operation. Digits can be ordered to form sequences or strings of digits which are the vehicle of digital computation (Piccinini, 2012). Roughly speaking, digits are *discrete* states of a system. These discrete elements can be letters of a finite alphabet or numbers of a binary or non-binary code. For example, digits can be used to express numbers such as "345", where each digit is a numeral with a specific place within the series (Maley, 2011). Likewise, digits can be used to express the concatenated strings of zeros and ones (e.g. "0011010"), which constitute the instructions of a computer's machine language. It should be noted that digits are discrete states of a system inasmuch as they are *unambiguously distinguishable* by the digital operation under normal processing conditions. The operation can distinguish atomic digits from one another because digits are macroscopic states of a system whose type can be reliably identified (Piccinini & Bahar, 2013). Computing systems are organized so as to reliably *distinguish digit tokens of different types* and manipulate them according to their type (Piccinini, 2008). Take the following string of digits: "0001101". The digital operation that computes it manipulates a finite number of digit types (0 and 1 in this case) and a finite number of digit tokens (four occurrences of 0 and three occurrences of 1). The operation is able to recognize the different occurrences of 0 in this string because it classifies them as a token of the type 0.

In short, a huge (but finite) number of distinct tokens of digits correspond to a same digit type and a digital operation is sensitive to this. This characterization of a digital operation embodies what Haugeland calls "*positive procedure*" (Haugeland, 1981). Haugeland understands a digital device (which performs digital operations) to be one in which the tokens of a set of specified types can be written and read reliably and with absolute certainty. From his perspective, digital devices involve some form of writing and reading of various tokens of various types. The procedure to produce tokens given the type and determine the type of given tokens is positive, in the sense that it succeeds in its scopes absolutely and "without qualification". So, a positive procedure manipulates digits reliably ("with astonishing precision") since, under normal conditions, there are no possible errors in the recognition of the corresponding digit types (Haugeland, 1981, p. 214).

I should say that Haugeland´s positive procedure would rescue what is often thought to be a “digital operation” in classical cognitivism. Back to traditional roots of cognitivism, such as Turing´s machine proposal, Newell’s perceptual symbol processing hypothesis and Fodor´s computational theory of mind, digital operation would be understood in terms of manipulation of uninterpreted symbols (Turing, 1950; Newell, 1980; Fodor, 1994). Following these approaches, symbols are combined exclusively according to their formal/syntactic properties (such as shape) and these properties would be best understood as discrete properties of digits which are transformed in digital computation (Fodor, 2000). For sure, digital operations constitute positive procedures because combinations of symbols according to their formal properties need to reliably read with enough precision to recognize some discrete/formal properties over others. If symbols were not discrete entities computed positively, it would be difficult to see the mind as a computing device causally sensitive to the syntactic properties of the symbols (Fodor & Pylyshyn, 1995).

I believe Merge is a positive procedure in the following sense: it is a compositional digital operation that manipulates special kinds of digits called “lexical items”. Each lexical item is a digit (e.g. “Mary”) and a sequence of lexical items is a string of digits computed by Merge (e.g. “Mary loves books”). Assuming that the FL has a lexicon which specifies the items that enter into the computational system, lexical items would be the complex objects that are combined by Merge (Chomsky, 1995). For example, “Mary” is a lexical item that can be considered a complex object since it has different linguistic features (e.g. semantic features, +ANIMAL, or syntactic features, +NN, among others). This item is grouped with others by Merge (e.g. “Mary loves books”) respecting their inner linguistic features. It should be pointed that lexical items are digits because they are discrete entities unambiguously distinguished by Merge. In part, Merge is able to compositionally combine the lexical items because this operation identifies the type to which the lexical item tokens correspond. The success of Merge as a set formation rule is due to the fact that it recognizes different types and identifies different tokens of a same type. In short, if Merge was not a positive procedure, it would not be an effective operation for the formation of hierarchical structures.

3. The Brain is not Digital

This digital characterization of Merge makes the study of its biological reality problematic. In order to explain this idea, I will focus on computational neuroscience. This area of neuroscience is grounded on pioneering artificial intelligence work (such as McCulloch and Pitts, as well as Rosenblatt) and biophysics (such as Hodgkin and Huxley) (Kass, 2018). These proposals advanced a new domain of research in which computational models were proposed to explain neural activity and brain function at all levels of detail and abstraction, from subcellular biophysics to human behavior (Kass, 2018). The current models of computational neuroscience are fed by concrete biophysical data and it is commonly believed that computational neuroscience also offers methods for the analysis of neural data (Kass, 2018).

Most computational neuroscientists believe that the neural system performs computations (Piccinini, 2006). It is true that this needs to be established by more than just the existence of computational neuroscience. Developing computational models has not committed neuroscientists to the conclusion that brains compute, since any phenomena can be computationally modeled (Piccinini, 2006). However, according to Piccinini and Bahar, brains seem to perform generic computations. Generic computation is the processing of vehicles in accordance with rules that are sensitive to certain vehicle properties (Piccinini & Bahar, 2013). Generally speaking, vehicles are the variables that permit the transitions of states in the computational process and rules are just mappings from input to output that needn't be explicitly represented. Assuming that this broad sense of computation captures a plurality of uses of this notion in cognitive science, it can be said that generic computation is physical when the rules are sensitive to the physical properties of the vehicles computed.

As Sarpeshkar establishes, a physical system uses three physical resources to perform its computations: time, space and energy (Sarpeshkar, 1998). As such, generic physical computation is the processing of vehicles according to rules that are sensitive to vehicle properties such as time, space and energy. A neural system performs computations in the sense that it performs generic physical computations. Neural signals include the propagation of electric charges and the diffusion of chemical substances in which we can find different possible vehicles of computation. Accepting some plurality of neural vehicles, spikes, hormones and neurotransmitters have the physical properties (duration, location and some energy consumption) needed to be considered the vehicles of neural computation. I want to emphasize in this section that the problem is that this generic physical computation, as presented, would not be digital.

It is true that McCulloch and Pitts argued that brains perform digital computation (McCulloch & Pitts, 1943). In their account of cognitive phenomena, they proposed idealized neural networks that operate on sequences of discrete inputs in discrete time. These networks produce digital outputs from digital inputs by means of discrete intermediate steps. More recently, Mochizuki and Shinomoto have proposed a mathematical model (the Hidden Markov Model) which analyzes electrochemical brain data in terms of discrete signals (Mochizuki & Shinomoto, 2014). The application of this model to brain data extracted from macaque monkeys shows that certain areas (e.g. the lateral geniculate nucleus) process information with discrete states. O'Reilly has made claims along the same lines as Mochizuki and Shinomoto (O'Reilly, 2006). He has proposed a biologically-based computational model of high-level cognition which supports some digital properties of the prefrontal cortex.

Nevertheless, non-digital approaches to the study of the brain have been around since the 1980s (Arbib, 1983; Beim Graben et al., 2008; Spivey, 2007). Many models of neural computation are based on continuous functions which are hardly computed by digital operations, as will be explained in Section 5. Agreeing with the general spirit of these models, I will try to show that the brain is not digital, in the sense that it does not process information using digital operations. It is true that authors such as Piccinini and Bahar intended to defend this negative thesis (Piccinini & Bahar, 2013). However, in this section, I would like to explore two different arguments that support this idea. These arguments take into consideration the already introduced notion of Haugeland's "positive procedure" (Haugeland, 1981). I have already argued that a procedure is positive if it can reliably succeed in categorizing a given token as an instance of its intended type without error. In Haugeland's account, digital operations are the procedures of physical devices that certainly type tokens. This would be the first premise of the following argument:

(i) A digital operation is a positive procedure
(ii) A positive procedure (indirectly) requires "noise-free" conditions
(iii) Brains would not operate under "noise-free" conditions
(iv) Brains would not use digital operations.

What does it means that a positive procedure requires "noise free" conditions? Remember that what counts as success for a positive procedure is the fact that the procedure actually produces a token of a required type, and it correctly identifies the type of the token supplied. This success condition depends on the possibility that the operation could correctly "read" and "write" items without any interference (Haugeland, 1981). For instance, what counts for an inscription of the digit "1" to be of the type "1", is determined by what the operation locally recognizes of such inscription without any active presence of background conditions that can obstruct the operation. Thus, a positive procedure is very precise because the suitable conditions provide a "noise-free" environment. As Piccinini says "Digital operations […] either they are performed correctly, regardless of noise, or else they return incorrect results, in which case the system is said to malfunction" (2008, p. 32). But, what is "noise free" in this context? Take electromagnetic interference (EMI) as a usual form of electrical-noise pollution in digital computation, in which positive procedures are performed. Such devices are dramatically sensitive to these interferences and "Disastrous, if not annoying results, occur if a system, subsystem or component interferes with another through electromagnetic means" (Getz & Moeckel, 1996, p. 1). Focusing on internal noise, components and subsystems are sources of EMI which emanate from one single element or a combination of components (Getz & Moeckel, 1996). The problem of internal EMI is considered during the initial design phase because it would be costly and often ineffective patches and going back into redesign. Design has the target of maximizing "noise immunity" or "noise free" conditions which can be described as device´s ability to prevent noise in its input from being transferred to its output (Getz & Moeckel, 1996, p. 8). EMI control techniques that pursue "noise immunity" involve both hardware implementations and procedures, such as shielding, filtering, wiring, among others. For example, shielding is used to reduce the amount of electromagnetic radiation reaching a sensitive victim circuit. In brief, noise free conditions depend on these kinds of techniques to eliminate interferences.

Having said that, the brain does not process information under "noise-free" conditions. Brain is not a device designed to eliminate interferences such as EMI. Neural activity is usually conceived as the processing of information by a single neuron and, by extension, the firing activity of neural networks. Neuronal noise designates random influences on the transmembrane voltage of single neurons and networks. These influences come from spontaneous brain activity not triggered by any sensory stimulus. It has been suggested that at least 75% of the brain's energy consumption comes from the spontaneous firing of neocortical neurons in cortical microcircuits (Raichle & Mintun, 2006). This general noise can influence, either regularly or stochastically, the transmission and integration of neurons' signals (Le Bon-Jego & Yuste, 2007). This influence on neural signals includes the chemical pacemaker-like activity of neurotransmitters and hormones, among others. In short, the processing of the brain would be affected by the interference of an indefinite number of physical conditions identified as "noise". Circuits are exposed to this kind of interference, which is not previously filtered. So, either the brain does not use positive procedures or it uses them under conditions of continuous interferences, in which case operations would never work correctly. This last scenario cannot be the case since the brain seems to process information correctly even with the presence of higher levels of spontaneous activity (González-Villar et al., 2017).

The second argument against the digital nature of the brain begins with the same premise as the argument above, but it exploits another characteristic of positive procedure:

(i) A digital operation is a positive procedure.
(ii A positive procedure (indirectly) requires that no token in fact be a token of more than one type.
(iii) We are not able to affirm that a neural spike is a token of just one type.
(iv) The brain would not proceed digitally.

Premise (ii) depends on the idea that digital operations individuate states that fall into the type-token distinction. As Piccinini (2008) says "…a string of digits is an ordered sequence of discrete elements of finitely many types, where each type is individuated…" (p. 34). Elsewhere, he adds that "programming a computer requires specifying how it must respond to different strings of digits by specifying how it must respond to different types of digit…" (p. 39). Despite the fact that it is difficult to clarify the difference between types and tokens of digits, it would be useful to follow Schneider (2011) in this respect. According to her, tokens of symbols/digits are physical states or "…patterns of energy (…) that fall into symbol types" (p. 120). Following Piccinini (2008), tokens in digital computation can be interpreted as "… physical digits that a computer can store" (p. 44) which are limited by the size of its physical memory. In contrast, types are thought to pertain in an abstract level. In this sense, types and tokens are not at the same symbolic level. While type entities would be computational symbols, token entities would be concrete particular physical states that fall under these abstract types (Wetzel, 2006).

It is well known that Haugeland (1989) introduces the concepts of "type" and "token" in digital computation by analogy to chess pieces. He underscores that pieces of the same type must function in the same way within a program, since interchanging them makes no computational difference (Schneider, 2011). The idea that any token can be substituted by another token of the same type in an operation without changing the computation is related to premise (ii) of the argument. Tokens of a same type are freely interchangeable, given that tokens pertain exclusively to one single type. If a token were related with more than one type, then substitution would not preserve the output of computations. According to premise (ii), in digital devices, types are "disjoint" in the sense that operations do not manipulate tokens that would be related to different types (Haugeland, 1981). For any candidate token, the positive procedure determines that the candidate is type "0" but it is not type "1". No token is ever equivocal between two distinct types. If that were the case, this ambiguity would conspire against the efficiency and certainty of the positive procedure. If there were indefinitely many types to which any given token might belong, the digital device would not do its job reliably, as it is supposed to do.

Once again, the brain seems to lack this property of digital devices. To analyze this, instead of discussing tokens of digits, I will focus on neural spikes. McCulloch and Pitts were the first authors to treat neural spikes mathematically as digits (McCulloch & Pitts, 1943). They conceived spikes as idealized all-or-none events (they either occur or they do not) unambiguously distinguishable. So, what constitutes a neural spike? Neurons have the ability to propagate signals rapidly over large distances. They do so by generating characteristic electrical pulses or spikes that activate the synapse of the neighboring neurons. These pulses are supposed to represent stimuli such as light, sound intensity, motor action, direction of the arm movement, etc. A spike is a short-lasting electro-chemical phenomenon that starts in the axon and crosses it just to arrive to the terminal buttons. A spike is a consequence of a series of chemical changes in the axon membrane, changes that produce electrical activity in the neuron (Craver, 2007). To summarize, brain activity involves the presence of many spikes produced by one or thousands of neurons.

Neural spikes are physical entities characterized according to physical properties such as duration and energy consumption (Schneider, 2011). This article has introduced the idea that these are the kind of physical properties to which the rules of generic physical computation are sensitive, and that neural spikes could be considered the vehicles of neural computations. It is true that there are different neural entities at different scales of the brain that are candidates to being vehicles. However, I want to concentrate on neurons as computational units and on spikes as vehicles allowing information to be transformed and combined before it is converted into neural output.

The problem with these kinds of vehicles is that it is difficult to split them into a number of finite spike types. Consider energy consumption, for example, one of the physical properties of spikes. Neural spikes are electro-chemical entities that demand certain energy. To the best of my knowledge, spikes are not classified according to the amount of energy needed for spiking activity. It is not true that, for instance, spike type 1 demands *X* amount of energy while spike type 2 demands *Y* amount of energy. Now, consider spike duration, another physical property of spikes. As it is assumed in the literature, the individuation of single spikes depends on the temporal occurrence of them. McCulloch and Pitts proposed to divide spikes into time intervals whose length was equal to synaptic delay. However, this is an idealization and real neural firing occurs with high variability in time due to a large number of factors in the cellular environment (Piccinini & Bahar, 2013). Spikes occur stochastically, in the sense that they do not occur within fixed time intervals. So, the same sample space for spiking events during a time interval is uncountable infinite. This means that a rule of neural computation would be sensitive to the temporal occurrence of spikes, but it would not individuate them according to certain types.

All of this frustrates the possibility of determining if a token belongs to one single type, as would be the case with a digital operation. If it is difficult to split spikes into a number of finite types, it is also difficult to determine if a token is related to one or more types. Digital operations manipulate digits which belong univocally to finitely many types. However, to classify tokens of spikes into types, first one should individuate them according to the presence or absence of the single token.

4. First Option for Merge

One consequence of the discussion presented in Sections 2 and 3 would be that Merge does not have any neural correlate. If Merge is conceived as a digital operation and the brain is not digital, then it is easy to conclude that Merge is a combinatorial operation proposed in theoretical linguistics that does not have any brain basis. Yet, this option is methodologically problematic in the context of biolinguistics. This is because this thesis is unconcerned with biolinguistic considerations. It conflicts with two main methodological assumptions accepted in this domain. The first assumption comes from what is called "*the minimalist program*" (MP). The MP is the latest development that continues the trend in generative grammar that began with Chomsky (Marantz, 1995). Generative grammar considers that natural language syntax is expressible by grammatical models endowed with recursive procedures, since natural languages involve recursive generative functions (Chomsky, 2002). Various kinds of generative procedures have been explored for decades. The MP proposed that Merge is the basic recursive procedure involved in syntax. As was explained, this operation recursively strings together two elements forming a third which is a projection of one on the other two. According to the MP, this generative procedure is basically a "monotonic composition of atomic elements" (Chomsky, 1995). Strictly speaking, the MP is not a theory but a research program that falls within the bounds of normal science (Chomsky, 2014). It is true that biolinguistics is independent of the minimalist program inasmuch as many of its questions can be addressed outside of the minimalist context (Boeckx & Grohmann, 2007). However, biolinguistics and minimalism have experienced a graceful methodological integration. They are related because Chomsky, initiating the MP, provided the most important sources for biolinguistics (Boeckx, 2013).

The MP seeks to approach the problem of determining the character of the FL from the "bottom up" focusing on how little can be attributed to Universal Grammar (Chomsky, 2007). In earlier approaches, it appeared that the design of language had to be highly articulated with many levels of representations and numerous intrinsically linguistic principles (Hornstein et al., 2005). But contrary to the idea that the FL must be reached with a complex and substantially unique structure, the "bottom-up" approach supports a more austere architecture of language. Indeed, the motivation for this bottom-up approach is substantially biological. It takes seriously the idea that the FL is, ultimately, a cognitive organ in which linguistics makes contact with biology (Yang, 2010). The bottom-up approach constitutes the kind of methodological considerations used to study organic systems (Chomsky, 2007). It is the general methodology related to the inquiry of biological objects comparable to the visual or immune systems, and other subcomponents of an organism.

The development of language, considered as any other biological system, involves three factors: (1) the genetic factors, which interpret part of the environment as linguistic experience, (2) the experience, which permits variations in a range of possibilities, and (3) principles not specific to the FL (Chomsky, 2007). The study of the FL in relation to these factors is engaged with "not entirely well-defined" claims such as "less is better than more" or "minimal search is better than deeper search" (Chomsky, 2014, p. 5). As with any domain of natural inquiry, proposals in linguistics are evaluated along dimensions such as parsimony and simplicity (Hornstein et al., 2005). The idea is to design a language system which has few and uncomplicated theoretical entities. Merge would be among these entities.

In its most elementary form, the computational system of language consists solely of the most efficient computational operation to interface with other components of the mind (Boeckx, 2012). Merge would be the only procedure to compute the output expressions of I-language. In the simplest case, Merge-based-systems are compositional systems that combine lexical items (Jackendoff, 2011). It is true that in a Merge-based-system it is not enough to capture all the facts about natural language. The minimalist claim is that, in order to kick-start all other linguistic operations, "all you need is Merge" (Boeckx, 2012, p. 322).

The second assumption that (jointly with the previous claims about minimalism) clashes with the thesis that Merge does not have any neural correlate is what I call "*the interactive levels of explanation*". Biolinguistics is essentially an inter-field research enterprise. It articulates developments from areas such as theoretical linguistics, psycholinguistics, neurolinguistics, cognitive ethology and genetics, among others. From a biolinguistic perspective, the properties of the FL cannot be studied by linguistics in isolation. The idea that cognitive capacity is the subject of different fields is intrinsically related to the idea that any cognitive capacity is subject to different levels of explanation (Marr, 1982). Fields and levels of explanation are not the same but, once a pluralism of fields is accepted, a pluralism of levels of explanation can also be accepted. In the case of biolinguistics, it can be identified as two main levels of explanation: the cognitive level and the neurobiological level of explanation.

Theoretical linguistics belongs to the cognitive level of explanation and neurosciences belong to the neurobiological level. This distinction of levels of explanation is not the same as that proposed by Marr in the case of vision (Marr, 1982). This author presented three levels usually organized in the following hierarchical order: computational, algorithmic and implementational levels. He considered the computational level to be the highest one in which cognitive scientists analyze, in very abstract terms, the particular type of task performed by the system (Bermúdez, 2014). Such a level constitutes a mathematical specification of what is being computed and why. It outlines the mappings from one type of information into another regarding a problem to be solved. Marr himself pointed out that his theory of computation was rather similar to Chomsky´s competence. At algorithmic level, it aims to work on how the mapping function studied at the higher level is processed in real time. Lastly, implementational level would be the one in which computational operations are implemented by physical mechanisms. A mechanism can be thought of as an organized structure that executes some function or produces some phenomena in virtue of containing a set of constituent parts or entities that are organized so that they interact with one another and carry out their characteristic operations and processes (Machamer, Darden & Craver 2000).

Let me focus on the implementational level which describes the physical realization of the mathematical mapping function that is being computed. According to Marr, this level does not incorporate computational notions. Following him, mechanisms implement computations as organized structures containing various constituent entities. In contrast to this, biolinguistics understands the neurological or implementational level as intrinsically computational. For instance, Carandini and Heeger (2012) proposed that many neural response properties can be understood in terms of canonical neural computations such as normalization, recurrent amplification, linear filtering or exponentiation. These are standard computational modules that apply the same operations in a variety of contexts. The key point is that these authors consider the nervous system as intrinsically computational. Computational neuroscientists operate with their own mathematical tool without committing themselves to implementional intentions. Even further, both cognitive and neurobiological levels are supposed to be computational given that the cognitive level presents computational primitives while the neurobiological level presents neural computations (Sarpeshkar, 1998). Both levels articulate computational notions, something that could not be affirmed in the case of Marr's approach.

One of the fundamental questions, in this case, would be how to integrate these different levels of explanation. The use of different levels gives rise to the question on how to relate the components at play in each level. Considering biolinguistics, the question is about the connections between current brain/language research. Following

Poeppel and Embick (2005), there are two different options to understand the interdisciplinary study of language and the brain. One possibility is that the study of the brain reveals aspects of the structure of linguistic knowledge while the other possibility is that language can be used to investigate the nature of computation in the brain. The first possibility is a "bottom-up" strategy whereas the second is a "top-down" strategy (Bermúdez, 2014). According to the "bottom-up" strategy, the neurobiological level of explanation guides the cognitive level, in contrast to the "top-down" strategy, in which the cognitive level informs the neurobiological explanation. In either case, there is a tacit assumption that "combined investigation promises to generate progress" in the study of language (Poeppel & Embick, 2005, p. 2). I agree with Poeppel and Embrick (2005) that the idea that neuroscience is in a position to inform linguistic theory, and vice versa, is clearly an open question. These two positions lack any obvious justification within scientific practice. However, the third option suggesting that these areas pursue an isolated program of research would not be attractive. Certainly, there is a serious risk of its being effectively *sui generis*.

It should be pointed out that in biolinguistics there is an interactive relation between levels of explanation, in the sense that "bottom-up" and "top-down" strategies come together. In Boeckx's words "It is true that one's view of language will determine the range of hypotheses one is willing to entertain when it comes to the biological bases of language, but the latter also depends on one´s view of biology" (Boeckx, 2012, p. 33). However, I would like to focus on the "top-down" strategy.

It proposes to take linguistic categories, such as Merge, seriously and use them to investigate how the brain computes language (Bilgrami & Rovane,, 2005). In my opinion this is a very promising strategy in order to integrate the brain/language research as encouraged by Poeppel and Embick (2005). It is true that it has deep problems that need to be solved. For instance, Poeppel and Embick identify what they call the "granularity mismatch problem" which states the difficulty to relate the fine-grained computational operations of certain linguistic theories jointly with the neuroscientific studies of language. Although this is a serious problem, I agree with Poeppel and Embick (2005) that it can be solved by postulating "computational operations that are at the appropriate level of abstraction" (5) being Merge one of these operations.

In sum, even accepting the "bottom-up" restrictions, the "top-down" strategy allows for the importance of Merge in the study of the neural substrate of language to be considered. Merge would be a computational notion proposed in the cognitive level of explanation that, in fact, guides the neurobiological inquiry. In conclusion, if it is assumed (i) that Merge is almost the only linguistic operation of the FL and (ii) that it leads to the development of certain neurobiological explanations then, for methodological reasons, it would be difficult to accept that this operation does not have any neural correlate.

These observations can be complemented with additional difficulties. The scenario presented in this section leads to two problematic options. According to the first one, the MP could take into account another operation instead of Merge. Setting aside Merge, the FL would be constituted by another recursive procedure which would guide the research on the neural basis of language. Some may think that minimalists are too obsessed with Merge and, given that generative grammars have proposed different recursive functions, it is time to end this obsession. However, to the best of my knowledge, no one in the MP has proposed any real candidate with the same recursive power of Merge. What kind of operation would replace Merge? What guarantees that this operation will be able to guide the neurophysiological research on language?

Another option would be to preserve Merge in the MP but accept that this procedure does not have a neural correlate. Poeppel and Embick (2005) maintain that the failure to detect computational linguistic notions in the brain does not necessarily imply that the notions are incorrect. Accepting this, Merge would be a correct minimalist operation that is not detected in brain activity. Nevertheless, biolinguistics considers the minimalist hypotheses that are supposed to drive naturalistic research (Lorenzo, 2013). If properties of language do not receive any naturalistic explanation, then “the biolinguistic approach comes down on the essentials” (Lorenzo, 2013). This methodological consideration should not be read as a mere stipulation. Chomsky understands that in biolinguistics, methodological considerations can often be reframed as empirical theses concerning organic systems (Chomsky, 2007). This means that the substantive hypothesis about language has to be reframed as an empirical option. The problem of Merge would be how to accommodate to this empirical possibility.

5. Second Option for Merge

Given, first, that Merge is characterized as a digital operation and, second, that the brain is not digital, then a possible conclusion would be that the brain realizes Merge but in a non-digital way. This section will explore the scope and difficulties of this thesis. For this purpose, the analysis will be divided into two main parts. Initially, the sense in which the brain would operate *not digitally* will be clarified, with the introduction of the idea that the brain could compute information according to analog operations. This will lead to an examination of how an *analog brain operation would realize a digital operation* such as Merge and some of the problems with this argument will be addressed there.

The idea that neural processes are *not digital* in the sense that they perform analogue computations is not new (Gerard, 1951; Rubel, 1985). In fact, analogue computations were proposed before the existence of digital computation and were used widely in the 1950s and 1960s (Piccinini, 2010). Analogue computation constitutes a broad notion which is often contrasted with digital computation. In this sense, "non-digital" and "analogue" would be synonymous. So, what would an analogue operation be? As mentioned above, a digital operation is a rule that maps inputs to outputs manipulating strings of digits. In contrast, an analogue operation is a rule that maps inputs to outputs manipulating continuous variables (Piccinini, 2010). Whereas the inputs and outputs of digital operations are strings of digits, the inputs and outputs of analogue operations are what mathematicians call "real variables" (Pour-El, 1974). Real variables are physical magnitudes that vary continuously within certain time intervals, taking any real value from a range of values. It is assumed that real variables fluctuate dynamically over time. Real variables are used in differential equations which are equations that mathematically presuppose that both the modeled physical systems as well as space-time are continuous. Algebraic differential equations have the form: $P(y, y^{t}1, y^{n}, \ldots, y^{t}n) = 0$, where P is a polynomial with integer coefficients and y is a function of x (Pour-El, 1974).

Unlike digits, real variables allude to non-discrete states of a system (Piccinini, 2010). If real variables vary continuously over time, then the occurrence of the different variables should not be understood as a sequence of all-or-none identifiable events. This characteristic of real variables shapes the kind of operation that manipulates them. Quoting Haugeland once again, an analog operation would be an "*approximation procedure*", that is, one which can "come close to perfect success" (Haugeland, 1981, p. 83). Approximation procedures are the antithesis of positive procedures in the sense that they admit a certain margin of error or degree of deviation from perfect success. According to Haugeland, the margin of error in approximation procedures is never zero, since perfect approximate procedures are impossible.

Why is the output of approximation procedures usually an approximation to the desired output? Why do analog operations have this low level of computational precision? In my opinion, the answer to these questions depends on the relation between the operation and the variables computed. Approximation procedures identify and manipulate real variables. Nevertheless, to the extent that these variables are non-discrete states of a system, the operation is not able to distinguish them unambiguously. As I presented, a digital operation distinguishes unambiguously tokens of digits as discrete states because the procedure can identify the type to which the token belongs. This is not the case for an analog operation, which manipulates occurrences of continuous variables without identifying any type to which they might belong. For instance, the *real variable 1* would be unambiguously distinguished from the *real variable 2* in the case of an operation that would identify the types of these tokens. However, an analog operation is sensitive to the instances of real variables but it is not sensitive to its possible types.

So, what does it mean to say that the brain, operating analogically, *realizes* Merge? I would like to limit the notion of realization by focusing on "Computational Brain Realization". According to this characterization of realization, both realized and realizer properties are computational. This notion is limited to the relation between the computational cognitive properties and computational psycho-chemical properties of the brain. It seems that realization is a metaphysical relation between the properties from different levels of organization (Weiskopf, 2011). The notion of "levels of organization" captures the idea that there are higher and lower ontological levels of realization. However, I want to connect the notion of "Computational Brain Realization" with the distinction of the levels of explanation as already mentioned in Section 4. There a distinction was made between the cognitive and neurobiological levels of explanation. With these levels in mind, it would not be controversial to consider that cognitive properties are posited by cognitive explanations, while neural properties are posited by neurobiological explanations. Taking into consideration the MP and computational neuroscience, both cognitive and neurobiological levels are computational. While MP posits computational notions in the cognitive level of explanation, computational neuroscience also posits computational notions in the neurobiological level of explanation. Cognitive scientists usually present their theories in computational terms and, in recent decades, many neuroscientists have started using computational notions to study neuronal activity (Piccinini, 2010). Thus, the properties posited by such theories would also be computational. Despite their differences, both kinds of properties are computational in a nontrivial sense because they are minimally presented as mappings from inputs to outputs in accordance with certain rules (Piccinini & Bahar, 2013).

Let us specify the nature of some computational cognitive properties introduced in "Computational Brain Realization". Some of these properties can be considered higher order physical properties. An example of this would be the syntax of mental representations as proposed by Fodor in the computational and representational theory of the mind (Fodor, 1987). It is true that biolinguistics is not intrinsically engaged with representationalism, but my aim in this part of the paper is to focus on the notion of "syntax" already presented by Fodor, just to study some of its properties. Syntax is a cognitive property since it belongs to mental representation, which is a notion posited in the cognitive level of explanation. It is also a computational property because it is a property of representational vehicles which are transformed by certain rules of transformation. Fodor considers that syntax is a higher order physical property because, at an abstract level, "it might determine the causes and the effects of its tokenings in much the same way that the geometry of a key determines which locks it will open" (Fodor, 1987, p. 19). Fodor thinks that syntactic structure would be an abstract feature of the shape of symbols and, because the shape is a potential determinant of their causal role, syntax would also be a determinant of its causal role.

I believe that Merge could be characterized as Fodor understands syntax of mental representations. This would characterize Merge with the appropriate level of abstraction demanded by Poeppel and Embick (2005) to relate linguistic operations with neurosciences. Merge would be a computational cognitive property realized by computational psycho-chemical properties of the brain emphasizing that Merge is a higher order physical property. As syntax would be a property of a physical property, such as the shape of symbols, Merge would be a property of some physical properties such as computational physico-chemical properties of the brain. Merge potentially and indirectly determines at a more abstract level some causal relations established in the brain. In this sense, Merge would be a second order physical property of the brain functionally presented. The idea that the brain realizes Merge not digitally would therefore be as follows: the brain instantiates Merge (as a cognitive property which constitutes an input-output mapping controlled by a digital operation) in virtue of the fact that it instantiates the same input-output mapping of Merge but controlled by an analog operation. This realization relation links the cognitive mapping of Merge with a neural mapping changing a digital operation for an analogical one.

Up to this point, this article has focused on clarifying in what sense the brain would realize Merge but not digitally. Whether the brain in fact realizes Merge is an empirical question that shall now be examined. Indeed, neural activity seems to have analog properties. Spikes, which are the most significant signals transmitted by neurons, have already been mentioned in the discussion of McCulloch and Pitts, who conceived of spikes as an all-or-none event of the nervous system (McCulloch & Pitts, 1943). However, the presence or absence of a spike at any time is far from an all-or-none event. 'Spike' refers to a rapid and fleeting change in the electrical potential difference across a neuron's membrane due to opening of voltage-gated sodium (Na^+) and potassium (K^+) channels (Craver, 2007). This potential difference, known as "membrane potential" (V_m), consists of a separation of charged ions on either side of the membrane. In a resting state, positive ions line up against the extracellular surface of the membrane and negative ions line up on the intracellular surface. In the spiking activity of the neuron, the membrane becomes fleetingly permeable to Na^+ and K^+ ions. This allows the ions to diffuse across the cell membrane, changing the V_m. The spike consists of a rapid rise in V_m to a maximum value followed by a rapid decline in V_m to values below the resting state. After that, the neuron enters into a refractory period where the cell is less excitable.

To be sure, this variation of the permeability of the membrane has analog properties since it can be characterized as the dynamical evolution of continuous variables in real time. Hodgkin and Huxley represented spiking activity as a continuous time-course of permeability changes as a function of V_m introducing the idea that the brain performs analog computations (Hodgkin and Huxley, 1952). Still, even accepting that neural processes have some aspects of analog computations, this is not enough to evaluate the possibility that the brain realizes Merge not digitally. To examine this last thesis, it would be necessary to find in the spiking activity that the input-output mapping with continuous variables corresponds to the input-output mapping of Merge. Isolated analog properties of the brain are not enough. They have to be presented as the neural realization of a digital operation such as Merge.

It should be recalled that a digital operation maps inputs to outputs manipulating strings of digits. In this mapping, the operation unifies different occurrences of tokens of digits under a finite number of types. In contrast, an analog operation is not able to unify the continuous variables manipulated under a finite set of types. This fundamental difference between digital and analog operations seems to be crucial for the case of the neural realization of Merge. An analog operation would effectively encode an input-output digital mapping in that it would manipulate real variables with unbounded precision. Given certain input-output digital computation, an analog operation would reflect it just in case it can assign a real value (e.g. *RV1*, *RV2*, *RVn*…) to each individual digit token (e.g. *token digit 1*, *token digit 2*, *token digit n*…). In other words, an analog operation has to deliver an output that is digitally produced according to type-token relations but, in this case, manipulating unbounded real variables considered independently.

In principle, a system is said to be continuous under a given mathematical description (Piccinini, 2008). This means that, at least abstractly speaking, the precision of the different real variables manipulated by an analog operation depends on the mathematical resources employed to design the device. It is true that a real variable can take any real number as a value and this gives some freedom for the system to achieve different levels of precision. However, analog operations of physical systems, such as the brain, inherit the limits proper of the physical device. In practice, a physical magnitude within a device can only take values within the bounds of the physical limits of the system. If some of the relevant physical magnitudes take values beyond certain bounds, then the system breaks down. For example, the values of the inputs and outputs of analog computers and their components must fall within certain limits, e.g. ±100 volts. (Piccinini, 2008). So, in the design of analog computers, engineers know the appropriate scale of continuous values that the physical device is able to compute. Nevertheless, this is not the case with the brain.

It is true that in computability theory, there are digital simulations of analog computations (Rubel, 1989). Besides, there are concrete computer machines, such as synthesizers, which are digital computers that simulate analog software. So, there are neither mathematical nor physical restrictions to simulating digitally analog functions. The problem I want to emphasize is not a conceptual impossibility of relating digital with analog operations. My point is an empirical one related to (i) the resources that we lack to study neural architecture and (ii) the complexity of this computing system which is the brain. Because neuroscience is in this stage of research and our brains are very complex systems with real variables, it is difficult to understand how a digital operation such as Merge could be analogically implemented in the brain. Currently, we have no idea which is the appropriate scale of continuous variables that the brain as a physical system is able to process. It is a problem of the current state of science that means we cannot determine if the brain has sufficient precision in the manipulation of real variables in order to implement digital operations such as Merge. Is the brain an analog system that performs computations over real variables with the sufficient precision to give rise to Merge? That empirical question has yet to be answered.

6. Conclusion

Biolinguistics faces two contrasting thesis when studying the neuronal bases of Merge. First, that Merge is a digital operation and, second, that the brain could not be understood as a digital computing system. In this essay I wanted to evaluate the consequences of these theses. I argued that, for methodological reasons, biolinguistics would not accept that Merge has any brain basis. Merge would be almost the only linguistic operation of the FL that guides most of the neurobiological explanations in biolinguistics. Moreover, I tried to show that the current state of the science is not able to establish if the brain, as an analog system, realizes Merge. It is an open question whether the brain has enough precision to implement a digital operation. In my opinion, either we accept the empirical difficulties of considering the brain as an analog system or we conceive the brain as a non-digital and non-analog processing system.

In facing these problems, there is a provisional strategy for studying the neural reality of Merge that can be advanced. First, treat this operation as a rule used in generic physical computation. As discussed, generic physical computation is the processing of vehicles according to rules that are sensitive to vehicle properties such as time, space and energy. Second, heterogeneous brain data about Merge must be taken into consideration. Functional neuroscience says that Merge seems to be localized in B44 and B45 (Broca's area) (Yusa, 2016; Schlesewsky & Bornkessel-Schlesewsky, 2013; Zaccarella & Friederici, 2015). From an electrophysiological perspective, Merge seems to correlate to the gamma and alpha range of oscillations characteristic of the thalamus (Boeckx & Benítez-Burraco, 2014). How to bring together this data while taking into consideration a minimal characterization of Merge as an operation of generic physical computation remains an open question.

References

Al-Mutairi, F. R. (2014). *The minimalist program: The nature and plausibility of Chomsky´s biolinguisitics*. Cambridge Press.

Arbib, M. (1983). *Brains, machines, and mathematics*. Springer.

Beim Graben, P., Pinotsis, D., Saddy, D., & Potthast, R. (2008). Language processing with dynamic fields. *Cognitive Neurodynamics, 2*(2), 79-88. https://doi.org/10.1007/s11571-008-9042-4

Bermúdez, J. L. (2014). *Cognitive science: An introduction to the science of the mind*. Cambridge University Press.

Bilgrami, A., & Rovane, C. (2005). Mind, language, and the limits of inquiry. In J. McGilvray (Ed.), *The Cambridge Companion to Chomsky* (pp. 181-203). Cambridge University Press.

Boeckx, C. (2009). The nature of merge: Consequences for language, mind and biology. In M. Piattelli-Palmarini, J. Uriagereka & P. Salaburu (Eds.), *A dialogue with Noam Chomsky in the Basque Country* (pp. 44-57). Oxford University Press.

Boeckx, C. (2012). The I-languages mosaic. In C. Boeckx, M. C. Horno-Cheliz & J.L. Mendivil-Giro (Eds.), *Language from a biological point of view: Current issues on biolinguistics* (pp. 23-51). Cambridge Scholars Publishing.

Boeckx, C. (2013). Biolinguistics: Facts, fiction, and forecast. *Biolinguistics*, *7*, 316-328.

Boeckx, C. (2013). Merge: Biolinguistic considerations. *English Linguistics*, *30*, 463-484.

Boeckx, C., & Benítez-Burraco, A. (2014). The shape of language-ready brain. *Frontiers in Psychology*, *5*, 282. https://doi.org/10.3389/fpsyg.2014.00282

Boeckx, C., & Grohmann, K. (2007). The biolinguistics manifesto. *Biolinguistics*, *1*, 1-8.

Boeckx, C., & Piattelli-Palmarini, M. (2005). Language as a natural object: Linguistics as a natural science. *The Linguistic Review*, *22*, 467-471.

Boolos, G. (1971). The iterative conception of set. *The Journal of Philosophy, 68*(8), 215-231. https://doi.org/10.2307/2025204

Carandini, M., & Heeger, D. J. (2012). Normalization as a canonical neural computation. *Nature Reviews Neuroscience, 13*, 51-62.

Chomsky, N. (1995). *The minimalist program*. The MIT Press.

Chomsky, N. (2002). *On nature and language*. Cambridge University Press.

Chomsky, N. (2007). Approaching UG from below. In U. Sauerland & H.-M. Gärtner (Eds.), *Interfaces + recursion = language?: Chomsky's minimalism and the view from semantics* (pp. 1-29). Mouton de Gruyter.

Chomsky, N. (2008). On phases. In R. Freidin, C. P. Otero, & M. L. Zubizarreta (Eds.), *Foundational issues of philosophical studies:*

Essays in honor of Jean-Roger Vergnaud (pp. 133-166). The MIT Press. https://10.7551/mitpress/9780262062787.003.0007

Chomsky, N. (2014). Minimal recursion: Exploring the prospects. In T. Roeper & M. Speas (Eds.), *Recursion: Complexity in cognition*. Springer. https://doi.org/10.1007/978-3-319-05086-7_1

Copeland, B. J. (1996). What is Computation? *Synthese*, *3*, 335-359.

Craver, C. (2007). *Explaining the brain: Mechanisms and the mosaic unity of neuroscience*. Oxford University Press.

De Mol, L. (2018). Turing Machines. In E. N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy* (Winter 2019 Edition). https://plato.stanford.edu/archives/win2019/entries/turing-machine/

Fodor, J. (1987). *Psychosemantics: The problem of meaning in the philosophy of mind*. The MIT Press.

Fodor, J.. & Pylyshyn, Z. W. (1995). Connectionism and cognitive architecture: A critical analysis. In C. Macdonald & G. Macdonald (Eds.), *Connectionism: Debates on psychological explanation*, vol. 2; Blackwell.

Fodor, J. (1994). *The elm and the expert.* The MIT Press.

Fodor, J. (2000). *The mind doesn´t work that way*. The MIT Press.

Fujita, K. (2017). On the parallel evolution of syntax and lexicon: A merge-only view. *Journal of Neurolinguistics*, *43*, 178-192.

Fukui, N. (2011). Merge and bare phrase sturcture. In C. Boeckx (Ed.), *The Oxford handbook of linguistic minimalism*. Oxford University Press. https://doi.org/10.1093/oxfordhb/9780199549368.013.0004

Gerard, R. W. (1951). Some of the problems concerning digital notions in the central nervous system. In H. Foerster, M. Mead & H. L. Teuber (Eds.), *Cybernetics: Circular causal and feedback mechanisms in biological and social systems. Transactions of the Seventh Conference, March 23-24, 1951* (pp. 11-574). Macy Foundation.

Getz, R., & Moeckel, B. (1996). Understanding and eliminating EMI in microcontroller applications. *National Semiconductor Corporation*, Application Note 1050.

González-Villar, A. J., Samartin-Veiga, N., Arias, M. & Carrillo-de-la-Peña, M. T. (2017). Increased neural noise and impaired brain synchronization in fibromyalgia patients during cognitive interference. *Scientific Reports, 7*(1), 1-8.

Haugeland, J. (1981). Analog and Analog. *Philosophical Topics*, *12*, 213-225.

Haugeland, J. (1989). *AI: The very idea*. The MIT Press.

Hodgkin, A. L., & Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology*, *117*, 500-544.

Hornstein, N., Nunes, J., & Grohmann K. (2005). *Understanding minimalism: An introduction to minimalist syntax*. Cambridge University Press

Hulst, H. G. van der (2010). Re recursion. In H. van der Hulst (Ed.), *Recursion and human language* (pp. 15-53). Mouton de Gruyter.

Jackendoff, R. (2011). What is the human language faculty? Two views. *Language*, *87*, 586-624.

Kass, R. E. (2018). Computational neuroscience: Mathematical and statistical perspectives. *Annual Review of Statistics and Its Application*, *5,* 183-214.

King, D. (1996). Is the human mind a Turing machine? *Synthese*, *108,* 379-389.

Kleene, S. C. (1952). Recursive predicates and quantifiers. In M. Davis (Ed.) (2004) *The undecidable: Basic papers on undecidable propositions, unsolvable problems and computable functions* (pp. 254-286). Dover Publications.

Le Bon-Jego, M., & Yuste, R. (2007). Persistently active, pacemaker-like neurons in neocortex. *Frontiers in Neuroscience*, *1*, 123–129.

Lobina, D. (2011). A running back and forth: A review of recursion and human language. *Biolinguistics, 5,* 151-169.

Lobina, D. (2014). Probing Recursion. *Cognitive processing*, *15*(4), 435-450.

Lobina, D. (2017). *Recursion: A Computational investigation into the representation and procesing of language.* Oxford University Press.

Lorenzo, G. (2013). *Biolingüística: La nueva síntesis*. Open Libra.

Machamer, P., Darden, L., & Craver, C. (2000). Thinking about Mechanisms. *Philosophy of Science, 67*, 1-25.

Maley, C. J. (2011). Analog and digital, continuous and discrete. *Philosophical Studies: An International Journal for Philosophy in the Analytic Tradition, 155*, 117-131.

Marantz, A. (1995). The minimalist program. In G. Webelhuth (Ed.) *Government and Binding Theory and the Minimalist Program: Principles and Parameters in Syntactic Theory* (pp. 351-382). Blackwell.

Marr, D. (1982). *Vision.* Freeman Press.

McCulloch, W. S., & Pitts, W. H. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*(4), 115-133.

Mochizuki, Y., & Shinomoto, S. (2014). Analog and digital codes in the brain. *Physical Review E*, *89,* 02275-1-02275-8.

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

O'Reilly, R. (2006). Biological based computational models of high-level cognition. *Science: New Series*, *314*, 91-94.

Piccinini, G. (2012). Computationalism. In E. Margolis, R. Samuels & S. P. Stich (Eds.), *The Oxford Handbook of Philosophy of Cognitite Science.* Oxford University Press. https://doi.org/10.1093/oxfordhb/9780195309799.013.0010

Piccinini, G. (2006). Computational explanation in neuroscience. *Synthese*, *153*, 343-353.

Piccinini, G. (2007). Some neural networks compute, others don't. *Neural Networks*, *21*, 311-321.

Piccinini, G. (2008). Computers. *Pacific Philosophical Quarterly*, *89*, 32-73.

Piccinini, G. (2010). The resilience of computationalism. *Philosophy of Science*, *77,* 852-861.

Piccinini, G., & Bahar, S. (2013). Neural computation and the computational theory of cognition. *Cognitive Science, 37*(3), 453-488.

Piccinini, G., & Scarantino, A. (2010). Computation vs. information processing: Why their differences matters to cognitive science. *Studies in History and Philosophy of Science*, *41,* 237-246.

Poeppel, D., & Embick, D. (2005). Defining the relation between linguistics and neuroscience. In A. Cutler (Ed.), *Twenty-first century psycholinguistics: Four cornerstones*, 1 (pp. 103-118). Lawrence Erlbaum Associates.

Pour-El, M. B. (1974). Abstract computability and its relation to the general purpose analog computer: Some connections between logic, differential equations and analog computers. *Transactions of the American Mathematical Society*, *199*, 1-28.

Raichle, M. E., & Mintun, M. A. (2006). Brain work and brain imaging. *Annual Review of Neuroscience*, *29,* 449–476.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain, *Psychological Review, 65*(6), 386-408.

Rubel, L. A. (1985). The brain as an analog computer. *Journal of theoretical neurobiology*, *4*(2), 73-81.

Rubel, L. (1989). Digital simulation of analog computation and Church´s thesis. *The Journal of Symbolic Logic, 54* (3), 1011-1017.

Sarpeshkar, R. (1998). Analog versus digital: Extrapolating from electronics to neurobiology. *Neural computation*, *10,* 1601-1638.

Schlesewsky, M., & Bornkessel-Schlesewsky, B. (2013). Computational primitive in syntax and possible brain correlates. In C. Boeckx & K. K. Grohmann (Eds.), *The Cambridge Handbook of Biolinguistics* (pp. 257-282). Cambridge University Press.

Schneider, S. (2011). *The language of thought: A new philosophical direction*. The MIT Press.

Spivey, M. (2007). *The continuity of mind*. Oxford University Press.

Turing, A. M. (1936). On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, *42*(1), 230–265.

Turing, A. M. (1950). Computing machinery and intelligence. *Mind, 236*: 433-460.

Weiskopf, D. (2011). The functional unity of special science kinds. *The British Journal for the Philosophy of Science*, *62*(2), 233-258.

Wetzel, L. (2006). Types and tokens. In E. N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy* (Fall 2018 ed.). https://plato.stanford.edu/archives/fall2018/entries/types-tokens/

Yang, Ch. (2010). Three factors in language variation. *Lingua*, *120*, 1160–1177.

Yusa, N. (2016). Syntax in the brain. In K. Fujita & C. Boeckx (Eds.), *Advances in biolinguistics: The human language faculty and its biological basis* (pp. 217-229). Routledge.

Zaccarella, E., & Friederici, D. G. (2015). Merge in the human brain: A sub-region based functional investigation in the left pars opercularis. *Frontiers in psychology*, *6*, 1818.

Received 21st March 2020; revised 22th May 2020; accepted 28th May 2020.