UNIVERSITÀ DEGLI STUDI DI TRENTO

Dipartimento di Psicologia e Scienze Cognitive

XXXII° Ciclo di Dottorato

On Computational Representations as Explanatory Tools for Intensional Reference

Relatore:

Prof.ssa Sara Dellantonio

Candidato:

Jodi Guazzini

Anno Accademico

2019/2020

«Just as in physics, the parts and states of the physical object are supposed to corresponds to those of the abstract concept. But in contrast to the situation in physics, we criticize *the material* part of the system when the correspondence breaks down»

- Minsky, M., *Computation: Finite and Infinite Machines*.

«Nam cum omnis ex re atque verbis constet oratio, neque verba sedem habere possunt, si rem subtraxeris, neque res lumen, si verba semoveris»

- Cicerone, *De Oratore*.

Table of Contents

Introduction

§1. The Renewal of interest in a Computational Framework

When Cognitivism has been instituted[1], a new research program raised on human mind, mental phenomena (behavior included, on the extent it depends on thought) and natural existence in general[2]. For human mind to be made subject of experimental practices, and to provide a scientifically reliable description of the processes underlying that mind, Cognitivism applied the theoretical and engineering tools from the crossing of five disciplines: neuroscience, logic, engineering of automated calculation, mathematical theory of function and communication engineering[3]. The concept of "mental representation" results from this intersection and it constitutes the core of the explanatory program of Cognitivism regarding mind and mental activity[4]. It must be always kept in mind that this concept is not at all equivalent to the *phenomenal* concept of "mental representation", that is the concept of a mind with *qualia* features and intensional contents, but the former aims to explain the latter, as I am going to show when I reconstruct the history of Cognitivism.

I claimed that Cognitivism is intrinsically bounded to the computational concept of "mental representation". As I am going to show where the relation between Cognitivism and Behaviorism is reconstructed from an historical perspective, there is a connection that involves the epistemic structure of Cognitivism. Cognitivism recollects many sciences, from psychology to neuroscience, from anthropology to – at least in some author view – chemistry and physics. It does so in an intelligible manner by distinguishing between levels of organization and levels of abstraction[5]. Different practices – that is, different epistemological norms, deontology and forms of description – correspond to these different disciplines, and neither cognitive psychology per se, nor pure neuroscience of physiology and behavior can provide a ground for coordinating these different practices within a unified, matching framework. The reason is that no discipline in the field can provide a

[1] I follow Gardner 1985 and adopt conventionally the year 1948, when the Pasadena Hixon Symposium was held.
[2] By "natural existence" I mean here not the way human beings are generated within a natural world, but the sum of activities and their results within human body.
[3] See Gardner 1985, pp.38-40; Shea 2018, pp.5-7. For a comprehensive history of the relation between computer science and Cognitivism, I followed mainly Boden 2006, over than Gardner 1985.
[4] Gardner 1985, pp.383-384.
[5] See Floridi 2008 for the distinction between "levels of abstraction" and "levels of organization", and for a definition of "level" in general too. As an example of the synthesis between these different levels, one can mention the "new mechanism" promoted by authors as Bechtel or Piccinini.

ground for building the levels on which each discipline can stay autonomous and legitimate, within a succession of growing complexity of organization, except computational neuroscience and/or physics, if one is willing to push the framework to complete reductionism, but this is considered just an ideal goal. Computational neuroscience can, since it provides an account in naturalistic[6] terms (compatible with the scientific method) for (A) the relation of "representing" – whatever it is, since there are many competing theories – and (B) the causality of mental representations.

Even if it has been criticized that Computationalism can account for the phenomenal features of mind[7], it is commonly agreed that it has the resources for explaining at least rational thought[8]. There are two widely shared criticism about the latter point too, namely Hubert Dreyfus' one in *What Computers Can't Do* (1972) and *What Computers Still Can't Do* (1992), and John Searle's argument in *Minds, Brains and Programs* (1980). Despite this past of criticisms, there is a renewal of interest in a computational theory of mind in many areas, since there is an increasing consensus that Computationalism is adequate to the dynamics of cerebral processes[9]. In philosophy of mind, there is a new confidence in the prospect that Computationalism will provide an explanation to mental phenomena as complete as possible[10], or at least it will contribute substantially to that[11]. Moreover, even if nowadays nonreductive physicalism defended by some version of the "multiple realizability argument" is the standard position in ontology within Anglo-American philosophy of mind[12], many important authors are working to reductionist theories within the computational theory of mind[13]; this is to say that Computationalism is a framework accepted for ontological reasons too and that it is still widely applied. Moreover, Computational studies and explanations are also becoming leading practices in both cognitive science and neuroscience[14].

---

[6] I understand "Naturalism" as in Nannini and Sandkhüler 2000, Kim 2003, Nannini 2007.

[7] By "phenomenal features of mind" I mean those features which imply an experience of "what-is-like". See for example Nagel 1974, Strawson 1984, Jackson 1986, Pitt 2004. Strawson and Dreyfus are probably the only authors that consider the issue of experience (understood as implying the necessity of interpreting sensibility through structures for it to make sense and being instructive) as not detachable from the problem of human activity. For a defense of this point and its contrast with the ongoing view in philosophy of mind, see Gallagher and Zahavi 2008.

[8] See for example Fodor 1975 or Johnson-Laird and Wason 1977.

[9] Churchland 1992, Bechtel 2008, Kandel et al. 2013, Rogers and McClelland 2014, Piccinini 2018.

[10] Churchland, Koch and Sejnowski 1990, O'Reilly and Munakata 2000. Gershman, Horvitz and Tenenbaum 2015.

[11] See for example Milkowski 2014.

[12] Kim 1998, p.8; Bickle 2010, Kim 2010, pp.122-124.

[13] As the reader can see from the texts in the previous notes.

[14] Over than Kandel et al 2013, cfr. Boden 2006, which includes also an history of computational neuroscience.

The outlined situation suggests that many authors in philosophy of mind and experimental sciences of mind are looking at Computationalism as the most promising prospect on explaining mind scientifically, even if there is a past of influential criticisms on the side of philosophy of mind, within which some authors (namely, Nagel, Jackson, Dreyfus, Strawson, Pitt, Gallagher and Zahavi) attempted to point out the importance of experience for human activities, but they are a minority. My purpose in this dissertation is to discuss whether or not Computationalism has the resources for explaining the concept of mental representation, in which Cognitivism grounds its promise of making mental activities objects of a scientific (experimental) enquiry.

During the enquiry proposed in the dissertation, demonstrative knowledge is of particular interest. "Demonstrative knowledge" is the act of thought that objectively describes external reality, locally (as descriptions of individual processes like in biology) or systematically (as for example in physics), and able to provide reasons for its being objective. There are multiple reasons for the interest in demonstrative knowledge:

1. demonstrative knowledge has been considered the area of human activities most compatible with computational explanations. Programs for problem solving are sometimes considered the earliest examples of computational explanation of human rational activity[15]. Thus, an enquiry on this topic can be considered quite close to the core of Computationalism;
2. the reason for which demonstrative knowledge is considered compatible with Computationalism, in my view, is that there is a presupposition according to which reasoning is independent of subjective experience because it is intersubjectively - hence impersonally – structured. Thus, a refutation of the idea that the experiential feature of mind is unnecessary in rational thought could be very effective and shed interesting lights on mind, knowledge and Computationalism;
3. the intersubjective, logical and methodic structure of demonstrative knowledge provides a substantial help in making defensible claims about it, or at the very least this allows to make those claims debatable;
4. "representation" is the most important concept of Cognitivism, and representation is a form of reference as well as knowledge is, since it can be considered as the demonstrative reference of thought to external reality. If one can show that

[15] This point of view has been promoted by Newell and Simon since Simon and Newell 1971, and this line of thought is also quoted as a step in their formulation of the concept "physical symbol system", which is specifically directed to the computational explanation of mind (in classical terms) – as it is mentioned in Newell 1980, p.136 and p.138.

knowledge actually is a form of representing in an informative sense, it is also possible to use an enquiry on knowledge as a representation to evaluate the explanatory prospects of Computationalism as a theory of mind, since knowledge would become an *explanandum* closely related to the core concept of Computationalism and Cognitivism, one traditionally conceived as explainable and meaningful within Computationalism.

§2. Two Epistemological Issues Concerning the Relation Between Specialized Use of "Representation" and Other Uses

In the previous paragraph I mentioned how crucial the concept of "representation" is, and I mentioned my interest in calling for attention to the role of the phenomenal, experiential features of mind regarding knowledge as a kind of representation. In this one, I want to circumscribe my research, by a brief discussion of some epistemological concerns proposed by William Ramsey in his *Representation Revised*. By consequence of Ramsey's discussion, I confine the general aim of my dissertation to two points: first, if it is legitimate to conceptualize the explanatory device "computational mental representation" as a "representation"; second, if "computational mental representation" can actually do that which it promises, i.e. explaining "phenomenal mental representation".

This is about the topic of the dissertation. The main reason for its structure is that "representation" is said in many senses, hence the first thing to do should be to clarify this concept, so that (A) it will be clear what it means to explain the phenomenal, mental representation by saying that something "computationally represents" something else in such a way that the latter accounts for the efficient causation of the former; (B) it will be clearer what the *explanandum* is. For these two points to be achieved, it is also necessary that (C) the former and the latter sets (since both qualitative and computational representations come in many flavors) get described in commensurable terms.

Without presupposing any of the forthcoming topics and discussions, let us begin by proposing an informal reflection, which affects epistemological issues I discuss through Ramsey's work. According to a famous metaphor, Wittgenstein compared language to a toolbox: «think of the tools in a toolbox: there is a hammer, pliers, a saw, a screwdriver, a rule, a glue-pot, nails and screws. The functions of words are as diverse as the functions of those objects. (And in both cases there are similarities)»[16]. "Representations" are like the objects in the toolbox and the functions of words in language: they come in many species and it is unclear if the homonymy is consistently grounded, completely ungrounded,

[16] Wittgenstein 1953/2009, §11, pp.8-9.

partially grounded on a family resemblance[17]. Consider in fact the following examples: the book *The Karamazov Brothers* represents in the same sense of the tragedy *The Karamazov Brothers*, once it is played in a playhouse? A proposition represents a situation in the same sense a picture represents the same situation? Are names representations, or they are not, and in the first case, do they represent in the same sense of propositions or whole books? And as a closing example, does a differential response of an organic or artificial system represent in one or more of the previous senses, or in some further sense, which in turn has its own divisions, as "representations" in natural languages, artificial languages, "fine arts" or performative arts have?

Ramsey 2006 argues that understanding what "representing" means is crucial for understanding the explanatory value of Computationalism, since the relation of representing something must be informative for the explanation via computational, mental representations to be informative in turn. Ramsey's starting observation is that a shift was ongoing within Computationalism itself between a "orthodox" paradigm (the so-called GOFAI[18], which I discuss in *Chapter Second*, based on the original work of Turing and its concrete realization in the Von Neumann's architecture), in which the concept of "computational representation" has originally been formulated under the idea of symbolic manipulation; and a connectionist paradigm, grounded on a new hardware architecture and more emphasis on the role of bodily internal organization, structure and capacities (embodied cognition). The two paradigms differ substantially, but they both stick to the claim that "internal representations" are necessary posited for explanations about mind and mental activity to be achieved[19].

Ramsey thinks that «the notion of representation has been transplanted from a paradigm where it had real explanatory value, into theories of mind where it doesn't really belong. Consequently, we have accounts that are characterized as "representational", but where the structures and states called representations are actually doing something else»[20]. His concern is that such a transition can cause some kind of "trivialization problem", that is the consequence that any physical system can be said to compute, if representing by computing only depends on mapping in general distal objects and events[21]. Moreover, such a general notion can make impossible even to conceive a non-representational account of mind,

[17] For "family resemblance", see again Wittgenstein 1953/2009, §67, pp.36-37.
[18] The abbreviation stands for "Good Old Fashion Artificial Intelligence".
[19] Ramsey 2007, pp.2-3.
[20] Ramsey 2007, p.3.
[21] See about the "trivialization problem" the related sections of Piccinini 2009, pp.4-5. See also Sprevak 2019.

which is probably a clue that a noninformative concept of "computational representation" has been applied.

Ramsey adds a further reason for considering his evaluation of the different forms of computational representation as an important enquiry, from which I want to start my research. Ramsey remarks that «to some degree, our current understanding of representation in cognitive science is in a state of disarray, without *any consensus on the different ways the notion is employed, on what distinguishes a representational theory from a non-representational one, or even on what something is supposed to be doing when it functions as a representation*»[22]. In fact, "representation" occurs in many contexts and for many different properties, and the formalized definitions provided for generalizing fail both in clarifying the property and in avoiding intuitive counterexamples[23]. After these issues have been acknowledged, Ramsey asks «why should we care if a given representational posit accords with our commonsense understanding of representation in the first place? If these are technical, scientific posits, which difference does it make whether the theorist uses the term "representation" to refer to things that behave in a manner sanctioned by intuition? Isn't really just the explanatory value of a theoretical posit that matters? And if so, isn't trivially true that cognitive systems use representations?»[24]. I think Ramsey's answer has some value as an *epistemological concern*: his point is that when one uses a notion for explaining, then it must be clear what this notion does *by being that which it is*[25]. For example, if someone says that "*x* explains *y* because both *represent* and *x* produces in a more intelligible way the same effects of *y*", then we are legitimate to ask the following (these questions are not explicitly asked by Ramsey, I summarize his reflections through them):

- What is the nexus signified by "to represent" which belongs to both? This question expresses the interest in the being informative of the application of some specific concept for explanatory purposes;
- How *x* represents? In other words, what does specifically (instead of "representing" in general) mean that *x* represents? This question expresses the interest in understanding the specific properties of the case at stake;

[22] Ramsey 2007, p.7. Italics added. Other authors, even if they do not agree with Ramsey's conclusion, acknowledge the accuracy of his analysis on how "mental representation" is used and described in cognitive science. See for example Tonneau 2011 and Sprevak 2011. For a division of the kinds of "computational representation" similar to that Ramsey proposes, see also Egan 2019.
[23] Ramsey 2007, p.9.
[24] Ramsey 2007, p.11.
[25] Ramsey 2007, pp.11-12.

- Why "representing" of *x* should explain "representing" of *y*? This question expresses the interest in checking if a proposed notion does the work it claims to do by specifying how it does it in a specific case.

Thus, Ramsey's remark is that when one applies a notion for explanatory purposes, it must be informative in the sense that *its application must provide a specific content to the explanation, i.e. it must distinguish one modality of explanation from others, within the same genus or between diverse relevant genera*. To be more specific, if Cognitivism says that *every* mental activity can be explained by appealing to different but *all computational* explanatory tools which are *all* classified as *representations*, and that they are explanatory *precisely because they are representations and representing is a core feature of mental activities*[26], then it must be sundry clear both what a representation is and if computational representations are actually representations, and if they are explanatory because they are representations; or at least, this should be a shared goal, an ideal to which direct our efforts, since of course many impairing difficulties stay in the middle of worthy enquiries.

This *does not* imply that computational representations must represent in the same sense of commonsense representations, it implies instead that *a commensurable sense of "representation" must be provided for evaluating the explanatory capacity of computational "mental representation" towards phenomenal "mental representation" according to the three questions outlined above*. On the light of Ramsey's reflections, "commensurable" has to be understood as "compatible to different uses", and as presupposing the will of avoiding the previous epistemological issues. It is an effort of finding the genus of many species, if there is any.

Unfortunately, the literature has not paid much attention to Ramsey's remarks. To the extent of my knowledge of the literature, there is no theory of representation which attempts to make commensurable the multiple senses of representing or the most relevant at least, hence intensional mental phenomena, computational representations and artifacts commonly called "representation" are all said to "represent", but they share only a generic acknowledgement of having the property of reference, in the sense of "standing for something else". The acknowledged property is then grounded on different ways of connecting something with a distal object, but theories about that are so different and

[26] This is not something that one will find in each cognitive science considered on its own. In fact, notice that I am always referring to *Cognitivism*, not to *cognitive sciences*. My problem is a problem on the line of principle. As I was saying in the previous paragraph, it is a matter of historical genesis and epistemological hierarchy that "computational representation" becomes the ground on which levels of organization of increasing complexity are built one on another. Of course, the accuracy of my claim is going to be documented, when I survey the relation between Behaviorism and Cognitivism in psychology.

specialized that it becomes unclear that which is acknowledged as "common", and it becomes difficult to understand if "representation" is a concept legitimately unified and what it means. In fact, the claim of having reference is too general to provide a satisfying answer to Ramsey's concern: reference is sometimes linguistic, causal, teleological, pragmatical (sensorimotor), ostensive and many more, hence researchers are left clueless on what reference as representing means.

I have been suggesting through Ramsey's reflections the idea that clarifying the concept of "representation" will make clearer also what explanations by computational representations are in Cognitivism (previous point A); what is the *explanandum*, that is the phenomenal sense of "mental representation" (previous point B); in the end, I exposed my claim that Ramsey's epistemological concerns call for the need of a transversal concept of "representation" (previous point C). Now I relate that which we have been surveying so far to my will of limiting the research to demonstrative knowledge, so that I can formulate a reasonable research question and plan its development.

§3. Framing a Research Question

In §1 I have suggested that contemporary Cognitive Science witnesses a renewed interest in a computational theory of mind, and that this is affecting philosophy of mind too, since Computationalism is again considered as a pivotal explanatory perspective on mental activity. I have also mentioned how important the concept of computational "mental representation" is within Cognitivism as a research program. In §2 then I proposed through Ramsey's analysis that, given the lack of clarity on the concept "representation", for epistemological reasons it is necessary to enhance our understanding of how "computational mental representation" does its explanatory job, by clarifying how this happens for this concept designating a "representation" qua representation and, eventually, by elaborating a concept of "representation" that makes commensurable the computational and the phenomenal senses of "representations", so that both terms get clarified and can be evaluated for the explanatory claim of the former on the latter. In this way, the previous three questions I extracted from Ramsey's line of reasoning can be discussed, since after that analysis it gets clarified what "to represent" means in both the *explanans* and the *explanandum*, and the former can be evaluated regarding its ability to translate a precise relation in the latter: "representing".

My work will proceed as follows. In *Chapter First* I discuss the genesis of Cognitivism from Behaviorism and the points of their opposition, hence I show that the concept of "representation" is actually as pivotal as I claim in the "epistemological architecture" of

Cognitivism, and that it is actually a concept that must be understood as computer science formulates it. From *Chapter Second*¸ in order to evaluate the explanatory prospects of Computationalism, I start a careful discussion on how different recipes of "computational representation" do their job and what they mean when they say "to represent". In *Chapter Second* the reader will find the discussion of the *classical model of computation*, that one directly indebted with Turing's concept of "Turing Machine" and its implementation in the Von Neumann's architecture. In C*hapter Third* I discuss in the same fashion the *connectionist model of computation*, which is closer to neuroscience and to contemporary Computationalism. In *Chapter Second* there is a discussion about the concepts of "information" and "information processing", which are common to both models of computation. That discussion is crucial to understand what a "symbol" is in computational sense, both regarding the classical model and the connectionist model of computation.

As I am going to show in the last chapter, the core of the explanatory power of Computationalism consists of replacing the declarative knowledge of a self-aware subject, that develops its content in an explicitly discursive forms, with the operational competence of a structure that acts properly under adequate conditions and performs *specific* operations (this remark makes sense because there are problems about what "to compute" means[27]). I am going to show also that this replacement implies in turn a theory in which interpretation of signs is grounded on their sensible and/or physical features, and consists of substituting ordered series of those signs with further series of the same kind and/or performing appropriate responses to such series.

[27] For a preliminary overview on the topic, I send to Piccinini 2008, pp.312-313; Piccinini 2009, pp.517 and following; Piccinini and Scarantino 2010.

Chapter First. An Historical Perspective on the Conceptual Role of Computational "Representation"

§1. The Methodological Instances Put Forward by Behaviorism

At this stage of the enquiry, the interest is in evaluating how Cognitivism constitutes a reply to some methodological issues raised by Behaviorists, so that I can show what the conceptual role of the concept "representation" is and what its origins are. In this way, I can document that which I said about representation being the very core of Cognitivism in §1, about its conceptual role in Cognitivism and its origin from computer science.

This historical reconstruction is the story of how a tension between two needs gets reconciled through the concept of "representation". At its starting point, psychology went through a non-homogeneous path for becoming an autonomous and empirical discipline. From the very beginning of its history[28], psychology experienced a tension between (1) granting the autonomy of its specific subject, something psychologists wanted to achieve by individuating a sufficient reason for "mind" (variously understood) to be considered a self-contained matter of studies; (2) the will of conforming to the standards of experimental method. These two points do not imply a tension by themselves: they came to imply one, because point 1 was mainly promoted following a phenomenological line of reflection; whereas point 2 was pursued through more mechanistic conceptions.

Given the purpose of the chapter, I begin this history of Cognitivism with the reconstruction of John Broadus Watson's (1878-1958) positions. For the historical and conceptual importance of his theories and epistemological remarks to be fully understood, it is necessary to consider that the pursuing of psychology as an autonomous science was itself divided into two approaches[29]:

- the attempt of asserting the autonomy of the discipline on the ground of some specific feature that belongs to mind. The most famous example is Franz Brentano's conception of "intentionality", who considered possible demonstrating through it a partial incompatibility between psychology and a purely quantitative approach[30];

[28] History of psychology conventionally begins with Wilhelm Wundt (1832-1920), one of the most prominent promoters of psychology as an empirical science. For a more detailed history of psychology, see Brysbaert and Rastle 2009.
[29] Poggi and Nicasi 2000, pp.129-132.
[30] Brentano 1874/1995, pp.14-15. Brentano stresses that mental phenomena differ from physical ones because they are not clues of some reality indifferent to how it presents itself to observers; they are instead the actual

- the attempt of promoting an experimental practice in psychology which is closer to the other sciences, centered on recollection and systematization of data in order to resolve circumscribed problems, commensurate to an experimental approach. In this way, psychology can be made a natural science, whose autonomy derives only from the being circumscribed of its object, that is also extended and articulated enough for it to deserve a separate enquiry. Even if with important differences, this was the stance of important authors such as Georg Elias Müller (1850-1934), Francis Galton (1822-1911), who introduced statistical methods in the experimental practice of psychology, Hermann Hebbinghaus (1856-1909), who replaced introspections of psychometry with inferential methods that are still central in contemporary practice.

The two methods differ in this: the former grounds the autonomy of psychology on definition and description of mind and mental activity; the latter grounds the autonomy of psychology on adoption of experimental practice. This difference needs to be stressed even if trivial, since it is through the existence of this twofold way of pursuing a scientific status for psychology that Behaviorism would have been able to satisfy at once both the first and the second need (previous points 1-2) of psychology, and to settle the second way of vindicating autonomy for psychology as the standard one. This was possible because Behaviorism created a genuine and robust experimental practice for psychology, and this is why it got a widespread consensus despite its radical theoretical positions.

American psychology inherited the tensions between the aspects listed at points 1-2, and it sided with Brentano's or Wundt's line of thought. William James promoted a psychology that acknowledges mental representations as events that are real on their own and as they present themselves. In his book, *Principles of Psychology*, there are arguments against reduction of psychology to physiology[31] and in favor of identity between consciousness and representation[32]. Moreover, James credited mental representations with being causally relevant regarding activities[33] that are sufficient and necessary conditions for

content of this showing, hence an actual, manifest reality is that which observers of mentality experience. This position and the property of "intentionality" indicated as the mark of the mental (see p.68) are sufficient premises for the conclusion that a purely quantitative approach is not exhaustive, and that a phenomenological-philosophical approach has a crucial role.

[31] James 1890/2007, pp.128-145. There is a criticism of a mechanistic explanation which necessarily implies a criticism to psychology as physiology, as the distinction of tasks outlined in the *Introduction* of that book confirms.

[32] James 1890/2007, pp.164-170.

[33] James 1890/2007, pp.326-356. In the chapter *Reasoning*, James maintains that «even if we are commonly unconscious that we infer at all» (p.327), nevertheless the activity of reasoning is a manipulation of the results that comes from the act of extrapolating features and defining ideas in a way useful for precise purposes. James explicitly indicated that his book defends a conception of reasoning and experience which he considers

acknowledging the possession of a mind[34].

The reason for which I mention James' theory in this historical reconstruction is that, despite James' intentions, his theory prepared the fortune of Behaviorism. James was indebted with Darwinism, in that: he thought that the role of consciousness must be understood by looking at how its activity helps the owner in the struggle for survival[35]. By performing an analysis under this premise, James' theory individuated as one of the main *functions* of consciousness that of *solving a task* established by a subject[36].

That being rooted into Darwinism started a paradox. James promoted this shift in the spirit of giving to mind the attention it deserves, against possible pretenses of physiologists while remaining within a materialist perspective: the point of this turn is that mind has to be considered according to the ends pursued by the dynamics of states of consciousness, not to the analysis of some underlying structure[37]. The problem is that Darwinism is mainly a theory of unconscious, material forces that accomplish results through organic causal chains; chains that individuals receives from natural history and preserve for contextual reasons. A "functionalist approach to psychology" then must be understood as the science of how consciousness contributes and participates to *activities of adaptation to the environment* within which the subject acts and with which it interacts, but these activities are such that the subject receives them independently from its choices, it exerts them without any intention, their content is not something it decides: the subject decides *how* to pursue a goal, *not the goal*. This has two important consequences.

First, consciousness is some part of an organic causal chain, and I am going to show that it is this idea of consciousness, as consisting of organic adaptive processes that cause actions, that which makes possible a "subversive" turn of psychology from the first perspective (Brentano and William's) to the other, since by claiming that consciousness must be understood through Darwinism as a function, then it becomes possible to conceive

---

the same as Locke's. See p.663:«for in truth I have done nothing more in the previous pages than to make a little more explicit the teaches of Locke's fourth book». In this case, since there are no unconscious representations in James' opinion, readers of James should acknowledge that, in his opinion, consciousness is the necessary condition for at least the terms of reasoning to be given, that is awareness is the necessary condition for at least the possession of the data (ideas) to be manipulated for fulfillment of purposes (this is reasoning).

[34] James 1890/2007, p.10: «*The pursuance of future ends and the choice of mean for their attainment, are thus the mark and criterion of presence of mentality*». Italics in the text.

[35] James 1890/2007, p.139. About Darwinism in James' work from historical perspective see Green 2009.

[36] See the already mentioned passage in James 1890/2007, p.10 in the light of James 1890/2007, p.140:«now the study of the phenomena of consciousness which we shall make throughout the rest of this book will show us that consciousness is at all times primarily a selecting agency». Consciousness is a selecting agency, and its activity of selecting means for future ends is the mark of its presence, *ergo*…

[37] See Angell 1907. See also the criticisms to the so-called "Automaton Theory" in the *Principles*. About a more general perspective on American Functionalism and Darwinism, see also Backe 2001 and Green 2009.

consciousness as a mean for ends that are impersonal, so that consciousness can be studied from the standpoint of what it *does* instead of what it *is*. Secondly and above all, when James posited the first premise, I think he realized a pivotal shift in Darwinism which (maybe) he didn't fully see, given the claims I reported about the independence of mentality. James claimed that consciousness is a "device" that selects means for ends, but since consciousness is itself a mean for adaptive purposes, then all that consciousness realizes is in turn a step of an adaptive process. According to this line of thought, James' theory leaves open the option of considering "adaptation" not only collective changes as the modification of habits and structures of whole species, or of numerous individuals within some, but also singular actions of singular individuals. Otherwise stated, singular "processes of selection" are adaptive actions that could be called "local", in the sense of being more tied to individual, circumscribed situations, if with "global adaptive process" one understands the complex processes Darwin described, which involve changes in whole populations of individuals, even if they involve the behavior of single members[38]. If there are those two consequences, hence there is a paradox: James promoted his theory with non-materialist intentions, whereas *this "local" sense is exactly the Behavioristic sense of adaptive process, and it is a materialist framework* that denies any attention to mind as such. These are the reasons for which I read the shift between Functional Psychology to Behaviorism as a "subversion".

Over than this theoretical continuity, the fortune of Behaviorism in psychology had two events as important conditions of possibility, that received a new synthesis in Watson's work[39]:

- the work of Ivan Petrovič Pavlov (1849-1936), that provides a model to conceptualize human behavior so that it becomes compatible with experimental practices. This model is composed by the model of reflex arc and the phenomenon of conditioning to a response given a stimulus. It was developed by Pavlov as a theory in neurophysiology. Its role in the rise of Behaviorism was to provide a tool for understanding "adaptive actions" in James' sense in a purely biological/neuroscientific way. Thus, Pavlov's idea allowed Behaviorism to conceive the adaptive acts of consciousness as behavioral acts grounded on biological events without consciousness (Watson's Behaviorism), without

[38] Costall 2004, pp.186-187 points out that some of Darwin's works deal with individual actions in a fairly close sense as the "local" one I stated. Nevertheless, I am not sure if his example cannot be considered "globally", as a preference in individual behavior that "global" adaptive processes carved into individuals.
[39] Poggi (2000), pp.513-520.

changing Functional Psychology. In this way, Behaviorism could attack Functionalism while standing on the same ground, and without breaking the tradition of psychology as it was so far;

- studies in animal psychology by Lloyd Morgan (1852-1936), Jacques Loeb (1859-1924), Edward Lee Thorndike (1874-1949) made psychology less anthropocentric. They provided experimental results that made legitimate considering mental operations traditionally attributed only to humans liable to natural explanation in unified, Darwinian terms, so that a prospect of completely natural explanation of human activities became open to Behaviorism. Thorndike especially gave a substantial contribution, since he showed that Pavlov's neuroscientific model fits high level operations (for example, learning), and he argued that Pavlov's model has a proper context in Darwinian adaptation theory[40], which is already present in James' work, as I showed. By consequence, it is legitimate to state that animal psychology opened a breach[41] into Functional Psychology through the Darwinian paradigm it had in common with Behaviorism. Animal psychology can be considered the "ground zero" of the shift from Functionalism to Behaviorism.

Now it is possible to fully appreciate the role of Behaviorism, thus let us proceed to its reconstruction.

In his *Psychology as Behaviorist Views It*, Watson justified the adoption of behavior as the object of psychology on the basis of epistemological reasons, and he explicitly connected his Behaviorism with the Darwinian theory of "adaptation" in a broad sense, which if I was right previously, James already prepared. Watson's main concern was to enable a robust experimental practice in psychology, and this is the ground on which removal of mind and consciousness is justified[42]. For epistemological issues to be overcome, Watson proposed to circumscribe the *observable* facts of interest for psychology to behavior only. Behavior is then defined in adaptive Darwinian terms, and conceived according to a Pavlovian model, which posits that to a given stimulus a certain answer necessarily follows, given a certain past conditioning[43]. In fact, Watson defined "stimulus" and "response" respectively

[40] Poggi 2000, pp.518-519.
[41] See also Angell 1907, p.71.
[42] Watson 1913, p.176: «due to a mistaken notion that its fields of facts are conscious phenomena and that introspection is the only method to ascertain these facts, it (psychology) has enmeshed itself in a series of speculative questions which, while fundamental to its present tenets, are not open to experimental treatment».
[43] Watson 1913, p.7: «The psychology which I should attempt to build up would take as a starting point, first, the observable fact that organisms, man and animal alike, do adjust themselves to their environment by means of hereditary and habit equipments. […] secondly, that certain stimuli lead the organisms to make the responses».

as «every object in the general environment or any change in the tissues themselves due to the physiological condition of the animal» and «anything the animal does»[44], then he added that "response" is that which animals do for alterations provoked by stimuli to affect them any longer[45]. Stimuli and responses then form ordered pairs, where an adjustment – the behavior that leads to remove the effect of stimuli[46] – follows a stimulus, and the history of being associated between a stimulus and a response is called "conditioning"[47]. This conception of "adjustment" is important, since here it is possible to see explicitly that shift from a "global" to a "local" sense of "adjustment to environment" and "adaptive acts" I have already pointed out.

Because of the conceptions stated above, Watson understood psychology as a science of adaptive acts that consist *only* of responses and, since he was understanding adaptive acts in a sense compatible with James', he was still Darwinian and Functionalist as his predecessors while he was grounding his model on Pavlov's work, but in a more radical and new fashion: James' psychology was Darwinian, but still left space for the idea of mind as an independent and "true per se" event; whereas Watson's move of taking only behavior into account, and doing so through a biological model, left behind precisely this idea while it remains consistent with the core explanatory concept of "adaptive act". Since Watson was at the same time within and outside the tradition to which Functional Psychology belonged, he could safely and correctly claim that «[…] *Behaviorism* is the only consistent and logical functionalism»[48].

The shift of perspective allowed Watson to credit psychology with a novel, higher position amongst sciences of life in a twofold way. First, Watson' Behaviorism posits psychology as an actor within a *transversal enquiry within life sciences*. Watson's Behaviorism made functional psychology and ethology (animal psychology) commensurable by appealing to a natural notion of function and behavior, so that he could propose a transversal study of animal and human strategies of behavioral adaptation[49], once that psychology would have abandoned every pretense on consciousness[50]; something that Watson regarded not only as

[44] Watson 1914/1931, pp.6-7.
[45] Watson 1914/1931, pp.13-14.
[46] Watson 1914/1931, p.14.
[47] Watson 1914/1931, p.13.
[48] Watson 1913, p.166. Italics in the text.
[49] Watson 1913, p.176: «The position is taken here that the behavior of man and the behavior of animals must be considered on the same plane; as being equally essential to a general understanding of behavior».
[50] Watson 1913, p.177: «This suggested elimination of states of consciousness as proper objects of investigation in themselves will remove the barrier from psychology which exists between it and the other sciences. The findings of psychology become the functional correlates of structure and lend themselves to explanation in physico-chemical terms».

convenient, but without consequences as well[51]. Secondly, Watson was able to define specific contributions of psychology to other life sciences. Watson's Behaviorism made psychology for behavior that which physics is for matter and biology for life: a general theory. Once that psychology could have stood on the same ground with natural sciences of living beings, its findings would constitute «the functional correlates of structure and lend themselves to explanation in physico-chemical terms»[52].

This is how psychology in Watson's terms became a "functional psychology" in a new sense. That conception made psychology transversal to the many species of living beings, and by defining stimuli and responses in adaptive terms, by taking advantage of James' widening of the concept "adaptive acts", Behaviorism could introduce itself as the general theory of functional correlates of biological structures and processes. In fact, «psychology as the behaviorist views it is a purely objective experimental branch of natural science. Its theoretical goal is the prediction and control of behavior. […] The behaviorist, in his efforts to get a unitary scheme of animal response, recognizes no dividing line between man and brute. The behavior of man, with all of its refinement and complexity, forms only a part of the behaviorist's total scheme of investigation»[53]. Notice that Watson retrieves a precise conception of scientific method: one centered on reproducibility of results and prediction of theoretical consequences.

Watson's main book, *Behaviorism* (1914), agrees with the previous work, but the nexus between Behaviorism and Darwinism became more explicit and radical with the reference to conditioning. Even if there are still defenses of the new theory on the ground of epistemological concerns, one can find harder criticisms to previous psychology in Positivistic terms[54]. In fact, Watson proposed a typically Darwinian definition of Behaviorism, «[…] a natural science that takes the whole field of human adjustment as its own»[55] and he explicitly grounds the explanatory model of the new science on Pavlov's work for the first time, even if he never acknowledges the origin of his explanatory model in the text: Watson only mentions Pavlov while discussing his famous experiments with dogs. Nevertheless, Watson's definition of "stimulus" and "response" is so close to

[51] Watson 1913, p.161: «one can assume either the presence or the absence of consciousness anywhere in the phylogenetic scale without affecting the problem of behavior by one jot or one little; and without influencing in any way the mode of experimental attack upon them».
[52] Watson 1913, p.177.
[53] Watson 1913, p.158. See also p.176: «the position is taken here that the behavior of man and the behavior of animals must be considered on the same plane; as being equally essential to a general understanding of behavior. It can dispense with consciousness in a psychological sense».
[54] Watson 1914/1931, p.6.
[55] Watson 1914/1931, p.11.

Pavlov's theory of conditioned response that Watson does not feel any need of translating Pavlov's terms, when he discusses the experiments.

I showed how early (Watson's) Behaviorism implies a theoretical transition from the first way of vindicating the autonomy of psychology to the second. This change continues through the works of Borrhus Friederic Skinner (1904-1990), but with some differences that are relevant for understanding how Behaviorism could recollect both ways of vindicating autonomy for psychology and establish a general experimental method for psychology. I am going to show that (A) Skinner's Behaviorism differs very few or even nothing regarding the reasons for a new psychology from Watson's, but (B) Skinner breaks the relation between Darwinism and Behaviorism, whereas he strengthened the relation between Functionalism and Behaviorism within a more complex theoretical and experimental practice.

When I discuss those differences, consider that there have been significant achievements in neuroscience (physiology at that times), which outlined a mechanistic model radically different from Watson's and that partially refuted Pavlov's model (see §3). Skinner did not include the new proposals in neurophysiology in his theories; instead he detached psychology from neuroscience. The new model in physiology emphasized the importance of functions as organizing and planning elements, so that they could be considered closer to Functionalism under this respect[56], even if the new physiology is nevertheless mechanical, not at all willing to consider consciousness as something causally significant as it presents itself. This can be of interest: it can be conjectured that Skinner's revision of Watson's model is indebted with the difficulties its model faced, over than with Skinner's own achievements.

(A) Skinner develops a circumscribed argument against a science of mind, which is the basis he uses to introduce Behaviorism. Skinner still applied epistemological arguments, but his ones differ from Watson's because Skinner's arguments are close to the problem of experimental practice, and he formulates them without any reference to the underlying biological structures or enquiries: his focus is entirely on psychology. Skinner's arguments will remain constant for his whole production, but I think the best explanation is the exposition in *About Behaviorism* (1974).

In the first chapter, Skinner argued that (1) if mental states are credited with being the sufficient reason for behavior, then they risk to provide a fallacious explanation, the kind

[56] Cfr. Poggi 2000, pp.523-524

post hoc, ergo propter hoc: anger causes an unpolite answer to a friend, but anger is exactly the state of body which *accompanies* the execution of the response "giving an unpolite answer"[57]; it is something that comes together with an event, not really an explanation, and it seems to have the same content. Moreover, the problem is not only applying a Humean concept of causality, (2) but (2_a) it is difficult to understand how a non-physical entity as mind could exert a causal efficacy on a physical existence as body[58]. (2_b) This affects the experimental practice: if researchers want to produce a certain behavior by manipulating environmental conditions, how can they account for the modification of environment – the only thing they can affect – causing mental states and those in turn causing behavior, if they posit mental states as intermediate entities?[59] Mental states seem unneeded elements, whose role is hard to explain once posited and uncomfortable to be manipulated.

(3) Skinner did not think that a reference to physiology can solve the issue[60]. The solution at stake consists of identifying mind with brain, so that the causal chain becomes completely physical and entirely observable, at least intuitively. (3a) In a passage from *About Behaviorism*, Skinner replied that «we cannot anticipate what a person will do by looking directly at his feelings *or* his nervous system, nor we can change the behavior by changing his mind *or* his brain»[61]. There is a twofold problem here: first, neither mind nor brain provides *observable* evidences of the happening of behavior; second, neither mind nor brain can be *manipulated* for experimental purposes. Thus, even if the hypothesis (as such it was at Skinner's time) of identity between mind and brain is true, this would not solve any problem, since the new *explanans* is hard to be both observed and manipulated. (3_b) Skinner makes a similar remark in a successive writing, *Can Psychology Be a Science of Mind?* (1990). Skinner writes that «if the mind is "what the brain does", the brain can be studied as any other organ is studied. Eventually, then, brain science should tell us what it means to construct a representation of reality, store a representation in memory, convert an intention into action, feel joy or sorrow, draw a logical conclusion, and so on. But does the brain initiate behavior as the mind or self is said to do? The brain is part of the body, and what it does is part of what the body does. What the brain does is part of what must be explained»[62]. Here the problem still is what it is observed by observing brain. Skinner's objection is that brain is not an *explanans* for a singular behavior only, since it is part of a

[57] Skinner 1974/1976, pp.10-11.
[58] Skinner 1974/1976, p.11.
[59] Skinner 1974/1976, pp.11.-12
[60] Skinner 1974/1976, p.12.
[61] Skinner 1974/1976, p.12. Italics in the text.
[62] Skinner 1990, p.1206.

wider causal chain than behavior in general, namely the bodily activities, that constitute a new *explanandum*: another topic, another enquiry, other problems. Thus, there is a second issue with physiology: it explains different things, and deals with different topics in a different manner.

In this article, it is possible to appreciate Skinner's skepticism about Watson's position on psychology as a functional explanation of underlying, physiological structures and activities: «physiology tells us how the body works; the sciences of variation and selection tell us why it is a body that works that way»[63]. Thus, Skinner ascribes different tasks for physiology and psychology: the former describes a causal chain, the latter as one of the "sciences of variation and selection" explains the genesis of this causal chain, what it is and how it comes to be. If this is the case, then biological sciences of life and psychology communicate through the Darwinian framework of evolution, but they answer different question; they can collaborate, but neither overlap nor can be considered as differing only in how they describe the same object.

These are the arguments that Skinner used for introducing Behaviorism, a science that can be understood in different ways. I discuss only «radical Behaviorism», since it seems Skinner's position, because the arguments in its favor in *About Behaviorism* are referred to Behaviorism in general in successive writings. As Watson already noticed, if psychology makes mind its objects, it risks getting involved into studies of an inaccessible object, so that a genuine experimental practice becomes unavailable. In the same way, Skinner argued that "radical Behaviorism" «[…] questions the nature of what is felt or observed and hence known. […] it raises the question of how much of one body one can actually observe». For this reason, researchers must confine themselves to the only observable factors: behavior and its dependence on environmental conditioning factors, synchronical or diachronical[64]. Thus, Skinner's point is that psychology should not deal with mind because there are multiple issues with it, which prevent psychology to have a robust experimental practice, i.e. mental states cannot be neither observed, nor manipulated, and they have a doubtful status as explanatory devices, since they seem tautological. These are all epistemological concerns, as I wanted to show. They share the same root of Watson's remarks: his problem was that mind is unobservable, as Skinner would have successively argued for, and that psychology cannot be an experimental science until it chases mind.

(B) We saw that Watson's answer to the problem consists of three main points: revising

[63] Skinner 1990, p.1208.
[64] Skinner 1974/1976, p.19.

the concepts of "adaptation" and "adaptive behavior", so that it shifted from a sense I called "global" – the original Darwinian one, at least in the *Origin of Species* – to a "local" sense; exploiting this revision for enabling an experimental practice in psychology, whose essence is manipulation of variables and reproducibility of effects; promoting the autonomy of psychology not on the ground of some features its object may possess, but of the scientific nature of its method, that legitimates a cooperation between psychology and other sciences of life. Now I am going to show that, on the other hand, Skinner (B_1) makes explicit some methodological aspects of revising the concepts of "adaptation" and "adaptive behavior" in the definition of "functional analysis", and (B_2) the same concept of "functional analysis" shows the reception of Watson's idea of experimental practice.

(B) In *Science and Human Behavior* (1953), Skinner expresses the will of making psychology rid of philosophical assumptions tied to the concepts of "cause" and "effect"; assumptions about how a cause should operate for the effect to be produced[65]. Skinner claims that his new definition is the one currently adopted within science, and soon it will replace any reference to "spontaneity of action" in each area of knowledge; an expression that Skinner regards as a mark of weakness within a theory[66]. The new definition is: «a "cause" becomes a "change in an independent variable" and an "effect" a "change in a dependent variable"»[67]. The reader can appreciate that this definition is purely formal, i.e. it does not mention that which a cause *is*, just its relation with an effect. Moreover, the attribution of the "cause-effect" relation is grounded on the capacity of *observing some change* in a correlate. As Skinner explains, «the new terms do not suggest how a cause causes its effect; they merely assert that different events tend to occur together in a certain order»[68]. Surprising enough, Skinner regards this as «important, but it is not crucial», whereas it is for at least two reasons. On the one hand, by rejecting any claim on individuating how causes operate, there is a transition from explanation as description of a causal link to explanation as description of covariances and how things happen; on the other hand, the new conception summarizes the opening of a wide range of new explanations that were unnoticed in Watson's schema, and that constitute the difference between the "stimulus-response" model (S-R model) and the "operant conditioning" model, as Skinner calls his own theory.

The former point definitively sets a diversity between Behaviorism and previous traditions

[65] Skinner 1953/2005, p.23.
[66] Skinner 1953/2005, p.48.
[67] Skinner 1953/2005, p.23.
[68] Skinner 1953/2005, p.23.

in psychology: structuralist psychology (psychometrics) and functionalist psychology claimed that psychological explanations deal with *why* mind operates and *what is* mental content; Skinner's Behaviorism prescribed to talk about *how* behavior proceeds and *how much* each factor affects the development of a process. The reader can see these questions are not just different, they are diverse in nature. Moreover, they make explicit the first consequence I draw from James' idea of considering consciousness according to its function in a Darwinian perspective: any enquiry on ends and definitions disappear from psychology and gets replaced with *operations and their development*. Thus, Skinner's science makes explicit the shift from considering mind for what it is and why it operates in a certain way to considering how mind operates and what it does.

The latter point is important because its formal character allows to consider as "stimuli" and "responses" a wider range of things than those considered by Watson. Skinner acknowledged that something in Watson's theory depicted «[…] behavior simply as a set of responses to stimuli, thus representing a person as an automaton, robot, puppet or machine»[69], but according to Skinner, this happened just because Watson was forced to count on Pavlov's model, not incompatible with the idea of mechanical causation[70]. Skinner avoided the same conclusion while remaining in Watson's model of Behaviorism. Skinner and Watson conceive psychology as a general science of behavior, whose aim is its prediction and control[71]. At the same time, they provide different understanding of how «the old "cause-and-effect connection" becomes a "functional relation"»[72]. Skinner extensively commented Pavlov's work in chapters four and five of his book, then he introduces some aspects of Pavlov's work that Watson did not receive, so that a new conception of behavior replaced Watson's. Skinner introduces new definitions[73], whose unity is the idea of probabilistic function in a mathematical sense:

- a "response" consists of "behavior", defined as something that *operates on an environment* to produce *consequences*. For this reason, behavior is now called "operant", since its value lies in the consequences of its being performed;
- the "reinforcer" is the factor that *arises the probability* of the operant, it is

[69] Skinner 1974/1976, p.4.
[70] Skinner 1974/1976, p.6.
[71] See Skinner 1953/2005, p.35: «we undertake to predict and control the behavior of the individual organism. This is our "dependent variable" – the effect for which we are to find the cause. Our "independent variables" – the causes of behavior – are the external conditions of which behavior is a function. Relations between the two – the "cause-and-effect relationships" in behavior – are the laws of a science. A synthesis of these laws expressed in quantitative terms yields a comprehensive picture of the organism as a behaving system».
[72] Skinner 1953/2005, p.23.
[73] Skinner 1953/2005, pp.; 60-68.

individuated *amongst the consequences of behavior*, and it can be *constant or contingent upon the operant*, and it can do its job by being either a pleasant or unpleasant consequence for the subject.

What are the factors a behavior is a function of? "External conditions" are all the relevant factors that affect the behavior and lie outside the organism[74], and they are distinguished in current conditioning factors, those present at a certain moment; and "environmental history"[75], that is the sum of past conditioning situations. Thus, Skinner's model divides the conditioning into three elements – namely, stimulus (external conditions), responses, reinforcer – and evaluates the history of conditioning in terms of how the reinforcer improves the connection between stimulus and responses as preferred but not necessary correlate variables within similar situations.

In the light of those definitions, it is possible to understand the differences between Skinner's Behaviorism and Watson's. Watson's model is a pure reflex arc: each stimulus gets paired with a response that it necessarily triggers in a direct relation, whose direction goes exclusively from the environment to the individual. Skinner's model instead conceives behavior as that variable whose value is probabilistically fixed by the interaction of many factors, weighted by the consequences in dealing with them and by the consequences those factors have on organisms. Since any behavior is at the same time an operant, i.e. it modifies the environment itself, since it is conceived simply as a variable that affects the probability of some others, since "dependent" and "independent" variable are just relative terms, Skinner's model allows to consider continuous chains of behavior, so that it is possible to understand events in the history of individuals as a long chain of behaviors and consequences, some dependent on the environment, other dependent on individuals. This is a more complex scenario than Watson's model. Since the picture that results is a connection of many variables that affect the probability of each other, it is possible to understand Skinner's model as a function in mathematical sense: a weighted connection of variables with certain values, whose combination realizes the behavior. Watson's model, instead, is closer to a classical, mechanical conception of causality, in which certain situations deterministically follow from others.

As I wanted to show, Darwin here disappears from the picture, or properly speaking he passes in the background, while Watson kept it in the front row: behavior is still a matter

[74] Skinner 1953/2005, p.31.
[75] See for this term Skinner 1953/2005, p.31; cfr. for the following explanation Skinner 1953/2005, pp.32-34.

of adaptation, that is, of providing responses to stimuli so that the needs from the latter are fulfilled by the former, but there is no more need of referring explicitly to evolution, since "environmental history" becomes synonymous with "conditioning history", both spontaneous and self-provided. At the same time, Skinner's Behaviorism enquiries mainly the *function of behavior* – both in the sense of the role of behavior in a given situation and in the sense of defining the values of behavior as a function – so that his theory is:

- *partially Darwinian*: Skinner did not conceive neurophysiology as something directly related to psychology, and he thought that evolution accounts for why conditioning works, but not for that which comes to be actually conditioned: as he puts it in *Science and Human Behavior*, «the evolutionary process can only provide a mechanism by which the individual will acquire responses to particular features of a given environment after they have been encountered»[76];
- *fully Functionalist and Watsonian*: James' prospect on consciousness consists of considering his evolutionary *function*, understood as what the utility of consciousness is for the survival of individuals. His answer was that consciousness selects proper answers in order to deal with ambiguous situations, but nevertheless has its true content and exerts its true function because it is that which it is: a phenomenal event with discursive and qualitative contents. Watson followed this line of thought and receives James' lesson about consciousness, but he emphasized the Darwinian root of James' though: the *function* of consciousness consists of providing adaptive answers to certain situations, but behavior evolutionistically considered is more than enough for this function to be accounted for. Skinner's idea of *behavior as a function* summarizes both: the function of consciousness – providing answers to ambiguous situations – can be described as a proper function that makes adaptation measurable, in which variables are weighted through the history of reinforcement, i.e. the positive or negative impact on individuals of consequences of former behavior in similar situations. Conditioning factors, behavior and its consequences as new stimuli on the individuals do the work of consciousness and fit an experimental method for describing adaptive acts in Watson-Darwinian sense; something that Skinner considered crucial for the scientific status of an enquiry. Skinner still conceived psychology as Watson – a science of prediction and control of behavior – and, as Watson, Skinner understood experimental practice in psychology as a matter of checking the correlation

[76] Skinner 1953/2005, p.55.

between stimuli and responses, but Watson conceives a simple application of Mill's inductive logic, whereas Skinner planned a more sophisticated model, closer to a mathematical and nomological conception of experimental science. Those are the methodological consequences already in Watson's change to the concept of "functional analysis of consciousness".

§2. Three Questions of Behaviorism to Psychology

In the previous paragraph, I showed how James' proposal of defining consciousness according to the function it may fulfill within a Darwinian perspective, together with Pavlov's neurophysiological model of reflex arc and the development of ethology as including equally human and animal behavior, were the factors that made possible the rise of Behaviorism, that in turn was able to fulfill the desiderata of psychology, namely providing strong arguments for its acknowledgment as an autonomous subject and the need of method adequate to scientific standards. I showed that Behaviorists were mainly concerned with epistemological issues and that they were able both to provide an autonomous, but not isolated, place for psychology amongst sciences of life, and to specify an experimental method for their enquiries. Thus, they satisfied those needs of psychology I mentioned at the beginning of this chapter, and they settled the tension between being autonomous and being a subject with an experimental and scientific method by a methodological and conceptual revolution that passed through a Darwinian understanding of the "function of consciousness"; function that was took in charge by behavior.

The previous exam allows to individuate three questions that Behaviorism asked to its contemporary and previous psychological traditions, which summarize the Behaviorists' objections to the introduction of mind as the object of a scientific enquiry with experimental methods:

I. *is mind an observable object?* Since private events are accessible within certain limits, or maybe inaccessible at all, even to the subjects that possess them; since they have the features of *post hoc, ergo propter hoc* fallacious explanations, it is preferable to posits as relevant variables, both dependent and independent, external and thus observable factors only: environmental causes of behavior, behavior itself and its consequences on the environment are outside the «world within one's skin», as Skinner once said, hence they are completely observable as any other physical event. By consequence, mind results to be out of reach, and introspection – the most shared method of psychology at Watson's times – unreliable;
II. *Since it has the features listed at the previous point, does mind fulfill the criteria of*

being object of experimental practice, i.e. the possibility of individuating and manipulating the causally relevant variables for the purposes of achieving predictions as confirmations of theories and reproducible results? The new definition of function and the choice of confining the enquiry to environmental and behavioral terms are consequences of a negative answer to this question and of the intention of keeping psychology within the boundaries of experimental practice, so that it can be a rigorous subject acknowledged as scientific;

III. *Is it necessary to admit a causal role for mental states as such?* Over than being inaccessible, mental states are vulnerable to the objection of causal exclusion[77]: according to Behaviorists, their proposed variables are a sufficient explanation[78], which has the advantage of avoiding epistemological issues and the impossibility of experimental practice.

These questions constitute as many points of disagreement with successive Cognitivist psychology, that would have to reply to those, since it thinks itself as able to restore mind as a legitimate object, compatible with those criteria for being scientific that Behaviorism had legitimately stressed. In the next paragraph, I briefly discuss the reasons for the decline of Behaviorism, whereas I postpone the discussion of the solutions Cognitivism offers through Computationalism to the last chapter, so that they can be analyzed in the light of a full analysis of the computationalist model of mental representation, which is the ground of the response.

§3. A New Model of Mechanism: Lashley's, Sherrington's and Tolman's Researches

Two complementary points are required for the transition from Behaviorism to Cognitivism: first, some refutation of the explanatory model of Behaviorists; second, proposals of alternative models. If one is willing to accept a progressive theory of science, he/she would correctly add that it is also required to extend the new theory to anomalies and phenomena left out from the previous position. All these three points were met by neurophysiology between the 20's and the 50's of the XX° century. Surprisingly enough, this means that neurophysiology was preparing and achieved a competing model of explanation almost twenty or thirty years before Skinner's writings, an author who felt legitimate to separate psychology from neurophysiology in a straightforward manner for

[77] I understand causal exclusion here as in Kim 2005, ch.2, §3, p.42.

[78] See Skinner 1977, p.9: «the mental apparatus studied by cognitive psychology is simply a rather crude version of contingencies of reinforcement and their effects». Skinner pushed the objection as far as to claim that mental life studied by Cognitivists is «a kind of Doppelgänger» of behavior and the factors that shape it, so that he pointed out its nature of "*ens multiplicatum praeter necessitatem*", thus to be eliminated, at least methodologically speaking.

methodological reason. Thus, it is possible to ask why neurophysiology was ignored by the leading author of American psychology, and why Behaviorism as we saw it in the previous paragraphs remained the most widespread practice until the 50's of the same century.

One possible explanation can be provided by the interpretation I advanced. Behaviorism was mainly an epistemological position joined with a demand for experimental practice of unquestionable scientific nature, hence neurophysiology may be considered still not enough advanced to provide an alternative experimental practice to Watson's and Skinner's proposals. In fact, even if Broca's and Wernicke's researches at the end of XIX° century were already available, the first neurophysiological, accurate localizations and the related engineering tools had still to come: they would have not until the 60's of the XX° century. Thus, even if the contributions discussed in this paragraph are sufficient to reject at least Watson's model of S-R conditioning, they are not developed enough to replace Watson's experimental methods for making mind a variable that can be manipulated and behavior a reproducible phenomenon. Under this respect, neurophysiology was still not an adequate counterproposal, but this was the core of the Behaviorist program, and in fact Skinner grounds his rejection of neurophysiology as psychology on the impossibility for the former to provide variables to the latter which can be manipulated. By consequence, Behaviorist psychology legitimately postpones or even ignores the translation of its findings and its explanations in neurophysiological terms. To sum up, the Behaviorist research program is impervious to criticisms to neurophysiology because these two fields are understood by psychologists (especially after Skinner, less by followers of Watson) as dealing with different topics and methods. This can be a sufficient explanation, given that my interest in this historical reconstruction lies mainly in the genesis of a theory more than in the account of the raise and fall of Behaviorism. Maybe things are even simpler: Margaret Boden, for example, states that physiology was deliberately ignored by psychologists after the 20's[79].

In the previous paragraphs, I also discussed Watson's and Skinner's perspectives on neurophysiology; now I am going to define better the relation between the two fields. Physiology was practiced together with anatomy and almost the totality of neuroscience until the 80's of XIX° century, and physiology was promoting a study of mind only in terms of seeking the localizations of the main mental functions: the interest was in individuating the anatomical groups responsible for each activity. This caused a submission of anatomy to physiology, in this: each process was sought according to definition and

[79] Boden 2006, p.264.

conceptualization of a given function, not by evaluating the anatomical structures per se[80]. Ramón y Cajal (1852-1934)'s work turned upside-down this perspective: Cajal showed that (1) the nervous system is composed of cells that differ regarding their morphology; and (2) those cells are not organized in continuous nets (reticular theory), as Camillo Golgi (1843-1926) claimed. In this way, (1) an anatomical principle was provided for enquiring the structure of the nervous system, to which the functions individuated could be correlated more precisely; moreover (2) the problem was posited of how nervous impulses could be transmitted between physically separate units[81].

The implications of Cajal's work were received in different ways. On the one hand, the Berliner school started by Hermann Von Helmholtz (1821-1894) practiced physiology through *in vitro* experiments, the analysis of each nervous fiber and its features, studied through the manipulation of chemical and physical variables. To sum up, this school judged a better approach the study of components for deducing the functioning of the totality. On the other hand, the French school, that was following Claude Bernard (1813-1878) and Michael Forster (1836-1907)'s theses, considered experiments on living beings, through vivisection and surgical operations, more significant for understanding how functions are realized[82].

Pavlov's teacher, Ivan Sechenov (1829-1905), studied in Berlin but applied their methods to the French program, for example in the enquiry about the mechanisms of sensation. In this way, Sechenov formulated the hypothesis that behavior is the quantitative result of the interaction between external stimulus and response of organism[83]: the direct ancestor of Pavlov's reflex. Pavlov widened Sechenov's theory by showing that stimuli can be associated with elements from heterogeneous classes, which become new stimuli and, according to Pavlov, the association consists of the formation of new patterns of activations. For example, in the famous case of dogs and bells, repeated presentations of food and ring of a bell produce a new connection between tympanum and saliva glands, so that the ring of a bell (learned stimulus) causes salivation, that would be a symptom of hunger in itself[84]. Since Pavlov thought that all those learned reflexes are coordinated by the brain, he focused his studies on *control of conscious behavior*. On its side, Behaviorism received Pavlov's study as the starting point for understanding the whole behavior as a variable that can be manipulated, thus psychology took Pavlov's work as a proof that

[80] Fantini 2000, pp.479-482.
[81] Fantini (2000), pp.482-483.
[82] Fantini (2000), pp.483-484.
[83] Fantini (2000), p.484.
[84] Fantini (2000), p.485.

behavior can be controlled, but it did not incorporate the idea that behavior depends on mechanisms that control responses, as I showed in the previous paragraphs. The idea of internal organization and of mechanisms that control behavior will be developed independently by those neuroscientists who, even if considered themselves as Behaviorists, provided the main experimental refutations to Behaviorism.

Karl Lashley (1890-1958) is the author who gave the most significant contribute both to refutation of Behaviorism and to construction of an alternative model in neuroscience. As Boden puts it, «on the one hand, he (Lashley) questioned the *neural reality* of the reflex arc. On the other, he denied its *theoretical adequacy* as an explanation of behavior»[85] In his *Brain Mechanisms and Intelligence: A Quantitative Study of the Injuries of the Brain* (1929), Lashley systematized his studies on learning of mice in mazes after they received injuries of brain. He documented that the loss of cerebral functions is proportional to the extension of damages in the dedicated areas. This was considered a contradiction to the model of reflex arc. In fact, it was plausible that the neurophysiological implementation of reflex arc consists of some pattern of consecutive and direct activations, instead the new results were suggesting that complex functions as learning result from interactions amongst different, specialized parts, as Lashley himself claimed[86]. The preferred model of Behaviorists has then become implausible. Gardner stresses that Lashley's work had further innovative aspects: «Lashley was also challenging two major dogmas of neurobehavioral analysis: the belief that the nervous system was in a state of inactivity most of the time, and the belief that isolated reflexes are activated only when specific forms of stimulation make their appearance»[87]. Unlike those "dogmas", Lashley's thesis for explaining his data was a system of top-down control which involves planning forthcoming action (which is another thing inconsistent with Behaviorism), and systems that are always active and hierarchically organized[88].

Even if he was causing big issues to Behaviorism, Karl Lashley's work started from the Behaviorists' standpoint, understood as a demand of more rigorous experimental practices, of a closer relation with neurophysiological interpretations of psychological data[89], and a legitimate restriction of the object of psychology to behavior[90]. Thus, Lashley considered Behaviorism mainly the project outlined by Watson, according to that which I explained

[85] Boden 2006, p.265.
[86] Boden 2006, p.265.
[87] Gardner 1985, p.13.
[88] Gardner 1985, p.13; Boden 2006, p.266.
[89] See Lashley 1923a, p.237.
[90] See Lashley 1923b, pp.348-349.

in §1 of this chapter. Despite this declaration of agreement, Lashley deeply reformed Behaviorism, but not in the way Gardner thinks, that is as a rupture[91]. According to Gardner, Lashley's talk at the Hixon Symposium fostered the idea that Behaviorism went wrong per se, that there was no viable answer in its paradigm, but Lashley's position is more complex and it may deserve some space to clarify that, since his position *was the first proposal of formulating the problem of behavior in terms of internal organization of the living being that acts*, which is the first step towards a theory of control in Cognitivist sense, that in turn is the context in which "computational mental representation" must be understood still nowadays, as I am going to show in the rest of the chapter. Lashley's talk at the Hixon Symposium stressed that another model is required for explaining behavior neurophysiologically, but he never said that a return to Brentano's or James' lines of thought is required. Lashley discusses the topic of Behaviorism extensively in *The Behaviorist Interpretation of Consciousness* (1923). He shows dissatisfaction for Pavlov's model and for the lack of attempts by Behaviorism of including mental phenomena in its paradigm, but that dissatisfaction comes from a lack of explanation for mental phenomena *in behavioral and neurophysiological terms*, nor in phenomenological nor in "functional" terms in the original perspective of James.

Lashley's main criticism is that Behaviorism did any attempt of making compatible to its system phenomena whose existence it admits, even if Behaviorism acknowledges they do not fit into its theses. For those reasons, Behaviorism left room for a parallel science, which at the same time it would like to forbid, but it did not provide any compelling reason for this[92]. Lashley summarizes the point as follows: «methodological behaviorism has all the faults of psychophysical parallelism plus that of intolerance»[93]. Nevertheless, he does not draw from this Gardner's conclusion that «behaviorist answer to questions of the human mind was no answer at all»; quite the opposite, he proposed that Behaviorism could become more coherent with the help of neurophysiology[94].

As a first step in order to demonstrate his thesis, Lashley stressed that both Behaviorists and "Dualists" (I guess he refers to Functionalists and Psychometrists, the two main schools of psychology at his days) accept that physics and biology can be theories that unify data from different areas and with different nature; thus, both of them must also

[91] See Gardner 1985, p.15: «those attending the Pasadena meeting were well acquainted with the scientific program of the behaviorists. And they shared an intuition – strongly bolstered by Lashley's tightly reasoned paper – that the behaviorist answer to questions of the human mind was no answer at all».
[92] Lashley 1923a, p.243.
[93] Lashley 1923a, p.243.
[94] Cfr. Lashley 1923a, pp.243-244.

accept a theory of knowledge compatible with those sciences, and as such that it allows them to include areas that they are not covering yet[95]. Then, he pointed out that dualistic theories attempt to justify themselves on empirical basis[96], hence the field where they fight each other is the mind-body problem, conceived as «a problem of the application of certain postulates and descriptive methods (those of the physical sciences) to certain specific data of knowledge (the so-called attributes or elements of consciousness)»[97]. Lashley designates the mind-body problem as the contended object because, in this way, he can conceive the quarrel with other psychologies as a methodological problem instead of an ontological one: the mind-body problem formulated in that way has any ontological commitment, it is just a question about the legitimacy of applying a method to specific, empirically individuated features. This is the reason for which he moves on with criticisms to the confusion in psychology between physical data, mental data, and above all the inaccuracy of the descriptions of consciousness provided by "dualists"[98].

Lashley divides the features of consciousness in features regarding content[99] and features regarding its organization[100]. His crucial premise is that introspection as a method implies a circular presupposition[101]: regarding the qualitative differentiation of contents, introspection provides contents as already differentiated, but there is no way of *accessing the process of differentiation*, hence there is no way of deciding whether contents are differentiated for their unique aspects or for their role within the process that produces them. Thus, introspection amounts to consider results as such, hence it is obvious that mental contents are described as immediate and simple data that are differentiated in an equally immediate way: this is that which they are at the end of the process, but only because they are considered from the final stage of the process[102]. The same problem applies to self-organization of content: Behaviorism as neurophysiology can provide a consistent account of its main features by discussing them in terms of selectivity and unity as properties of processes, namely some elements are excluded because they are irrelevant as arguments for specific processes, and processes have unity because they make some elements relevant for the outcome[103]; this is the reason for which Lashley concludes that

[95] Lashley 1923a, p.245.
[96] Lashley 1923a, p.246.
[97] Lashley 1923a, p.246.
[98] Lashley 1923a, pp.246-247.
[99] Listed at Lashley 1923a, pp.248-249.
[100] Listed at Lashley 1923a, pp.249-250. This section includes the points more interesting for the question of the internal organization of living beings and its relation with behavior.
[101] Cfr. Lashley 1923a, pp.251-253.
[102] See also Lashley 1923b, p.329.
[103] Lashley 1923a, pp.262-263.

self-ordering «is as inherent in our conception of material world as in consciousness, and irrelevant to the argument»[104].

Lashley indicates as the main outcome of his analysis of consciousness that «their (of conscious contents – my note) "peculiarly psychic" attributes of quality and reference are not intrinsic to them as self-existent elements, but can be defined only in terms of their relationship within the complex organization of which they are independent variables»[105]. The novelties introduced with Lashley's proposal at the level of neurophysiological mechanism are mainly three:

1. unlike classical mechanisms, the new model includes processes within which produced states act on previous states, the mechanism "reacts to its own reactions", as Lashley puts it[106]. This can be considered an intuition of the concept of "feedback", that will be central in the "cybernetic synthesis";
2. by consequence, the internal organization and history of the behaving subject shape responses[107];
3. Lashley's model implies coordination and integration between different areas and functions[108].

To sum up, Lashley's work contributed to the refutation of Behaviorism by challenging its underlying model, that implied a direct triggering of some response by a stimulus, in this: functions are stored in areas, they are not embodied in direct successions; the order of the process is not serial, since behavior is weighted according to habits, it does not just consist of habits, hence internal processes of the subject elaborate stimuli and plan responses. Over than this, Lashley contributed to elaborating a new model of mechanism. The revolutionary aspects of Lashley's proposal are foreshadowed in his paper *The Behaviorist Interpretation of Consciousness*, but they become fully articulated in his talk *The Problem of Serial Order in Behavior*.

Lashley cannot be credited with being the inventor of the cybernetic/connectionist model of information processing, even if he guessed some of its core principles – processes with

[104] Lashley 1923b, p.330.
[105] Lashley 1923a, p.261. See also p.269 and Lashley 1923b, p.336: «we have seen that awareness is defined only by the attributes of content and the reactions of our machine have all of the subjectively definable attributes of content».
[106] Lashley 1923b, p.333.
[107] According to Gardner, this is also a claim in Lashley's talk at the Hixon Symposium. Cfr. Gardner (1985), p.15:«Lashley concluded that the form precedes and determines specific behavior: rather than being imposed from without, organization emanates from within the organism». Lashley stresses the point repeatedly in *The Behavioristic Interpretation of Consciousness*. See Lashley 1923a, pp.264-267. See also Lashley 1923b, pp.333-335.
[108] Lashley 1923b, pp.332-333.

feedback, integration of stimuli from different areas and different processes for realizing different functions, control of responses to stimuli as a matter of internal organization – since he proposed mostly clever intuitions and sharp hypotheses, grounded on his experimental outcomes; moreover, his ideas were framed in biological terms, not mathematical. Instead, for the proper computational model of processes to be fully developed, tools from logic, mathematics and engineering were still missing, and they do not at all supervene accidentally over the principles of the model. Nevertheless, his contribute to the "cybernetic synthesis" is hard to be overestimated: he is the first to conceive the possibility of joining physiological variables (integration, procedural incompatibility, and the like) with the problem of control of processes in mechanical terms, and those are the fundamentals of the "cybernetic synthesis"[109]. Moreover, according to my reconstruction *Lashley makes a crucial move, that contributed to bring Behaviorism into Computationalism, since his work transferred the model S-R from overt behavior to internal functioning*, which will prove to be the pulsing heart of the "cybernetic synthesis".

For the sake of completeness, it must be considered that an important contribution to the idea of integration between functions and processes came from another neuroscientist, whose work is another objection to the Behavioristic model of reflex arc, Charles Sherrington (1874-1952). Sherrington was a member of the French-British school of neurophysiology, which developed Cajal's work. In 1906 Sherrington collected the results of his studies into a series of lectures held at Yale University, successively published as *Integrative Action of the Nervous System*. He showed that responses from reflexes could be inhibited or enhanced by intersections with stimuli from other circuits within organism, thus there is an active control of the organic system on incoming stimuli and a modulation of them in order to perform refined behaviors[110]. The very idea of "machine that passively adapt itself to environment" promoted by Behaviorism was challenged by those studies: Sherrington's work showed living beings as able to control themselves, that is able to formulate independently strategies for interactions with its environment. Moreover, Sherrington's work was challenging the idea that some direct causal chain is responsible for the most complex operations performed by living beings.

Psychology was also preparing competing alternatives to the S-R model, but in this case as well as in neuroscience the involved authors was still considering themselves as Behaviorists. Even if Behaviorism was the major research program during its days, it was

[109] Gardner (1985), pp.19-21.
[110] Cfr. Fantini (2000), pp.489-490.

neither monopolizing nor monolithic. The study of learning prepared conflicts that would have gradually caused a crisis within Behaviorism. According to Stefania Nicasi, «the debate on learning, and specifically on animal learning […] witnesses the opposition of two different tendencies: on the one hand, a mechanistic approach, grounded on the S-R model; on the other hand, a cognitive approach, not resistant to the principles of Gestalt psychology and the results from the researches of Kölher, neither to neuropsychological theories of brain as a totality»[111].

Edward Chace Tolman (1886-1959) provided the most significant data against Behaviorism amongst psychologists. Tolman repeated classical experiments on explorations of mazes by mice, but without inserting any element of reinforcement. Not only mice explored the mazes, but they could solve problems faster, if a previous exploration of the mazes was allowed, even if there was any reward, hence without any reinforcement[112]. Tolman explained his results according to Ralph Barton Perry (1876-1957)'s theses, who claimed that the concept of "purpose" is necessary for many psychological explanations, and that it is not possible to posit purposes without postulating the existence of believes as well[113]. Moreover, Tolman was also convinced that Gestalt psychology was correct regarding the necessity of positing organizational principles for some performances to be explained[114]. Thus, Tolman discussed his results explicitly as demonstrations that concepts as «goal-seeking (purpose)», «initial explanatory impulses (hunches)» and «final object adjustments (final cognitions)» are useful for studying animal learning. Nevertheless, Tolman was not against Behaviorism: he stressed that those concepts are defined «objectively and behavioristically, not "mentalistically"»[115].

Tolman's interest in the theories of Gestalt psychology is showed in how he explained his data. He saw the behavior of mice as resulting from the confirm of their expectations and from the organization of stimuli in goals that form "cognitive maps" or "Gestalten"[116]. This violates two central points of Behaviorism as a research program:

1. Behavioristic theories of learning do not allow that adaptive strategies – to which learning is reduced – are planned before being executed by consequence of the

[111] Nicasi 2000, p.1010. My translation, original text as follows: «il dibattito sull'apprendimento, e in particolare sull'apprendimento negli animali […] vede in sostanza la contrapposizione di due diverse linee di tendenza: da un lato un approccio meccanicistico fondato sul modello S-R; dall'altro un approccio cognitivo, non sordo ai principi gestaltici e ai risultati delle ricerche di Kölher, né alle teorie neuropsicologiche del cervello come totalità».
[112] See Tolman and Honzik 1930.
[113] Boden 2006, p.260.
[114] Boden 2006, pp.260-261.
[115] Tolman 1925, p.285.
[116] Boden 2006, p.261.

interaction with the environment. The opposite case implies, on the one hand, that a subject conceives its actions, but this would imply in turn that mind is causally relevant, at least as far as it is identified with its own contents; on the other hand, it implies also that there are structures that organize behavior which are not derived from environment, but are formed independently by the subject who elaborates the stimuli from that environment. Both of those consequences are unacceptable for orthodox Behaviorism, and both of them follow from Tolman's results and the way he accounted for them;

2. If there are not reactions without actions, there must not be activities without stimuli, given that behavior is a pure reaction according to Behavioristic psychologists. If this is the case, there cannot be activities for their own sake, since a stimulus was defined as some alteration that the living being aims to suppress and the sufficient reason of all his actions. The spontaneous exploration of mazes without rewards by Tolman's mice was exactly such an activity.

For the outcomes of this overview on history of neuroscience and Tolman's refutation of Behaviorism to be summarized, it can be said that the model of stimulus-response chain, from which Behaviorism derived its main explanatory model is related to the cybernetic synthesis through the history of neuroscience. We have seen that this model went into many changes. At least in the early formulation of Behaviorism outlined by Watson, the causal explanation in terms of stimulus-response chains were Pavlov's model of reflex arc, whereas Skinner's operant conditioning is more sophisticated, and maybe closer to some aspects of Lashley's model, which introduces substantial changes, but nevertheless sticks to some crucial aspects of Behaviorism: its conception of science, its concept of adaptation and adaptive function, its positing responsiveness as the only legitimate explanatory factor. Thus, Behaviorism and the original core of Cognitivism are not mutually stranger: some of the central principles of the latter come from the neuroscientific translation and improvement of the former. By consequence, Pavlov must be credited for showing or at least suggesting that neuroscience can account for the coordination of reactions performed by living organisms; an idea – both Pavlov's model and the one of organic existence – that Lashley and others were just trying to improve. In the same way, Gall and Lombroso are nowadays credited (see the first pages of Kandel et al. 2013 for example) for having suggested that neuroscience can account for moral features and high-level abilities of human beings.

It is interesting to notice that the authors I surveyed do not conceive themselves as

opponents of Behaviorism: as Stefania Nicasi explains in his quote, their approach differs substantially from that of Behaviorists, since they apply constructs that are incompatible with Behaviorism in various respects, but those authors' purpose was to improve Behaviorism, since they shared both the epistemological concerns expressed by Watson and Skinner, and the consequent restrictions regarding the object of psychology. This suggests that, even if their theories are incompatible regarding content, those authors believed to follow the same principles of Skinner and Watson, thus their theories are not supposed to violate them. Since we have seen that they actually do regarding the model of the processes which the new theories apply, unless we are willing to postulate that these authors are wrong in understanding their own theory, a reasonable explanation may be that they were sharing the previous epistemological concerns I outlined, and the consequent restrictions on the object of psychology. In fact, any of them was talking about mind as such.

At the same time, there are genuine, non-reducible changes that ground a neat distinction between Behaviorism and a new proposal, which is still at its early stages but whose features challenges directly the Behavioristic model of explanation. Lashley's and Sherrigton's researches posit three points that are going to lay the foundation for a new model of self-regulating mechanism, which the cybernetic synthesis of informatics, logic, mathematics, engineering and neuroscience will make theoretically describable and artificially realizable. In this way, theses on control and realization of behaviors was made experimentally testable and reproducible within the theory of control of input stimuli by more and more complex mechanisms.

Two remarks can be added. First, the first general idea of the problem of behavior as a problem of control within mechanisms was formulated by neuroscientists who conceived themselves as Behaviorists, whereas Behaviorists in psychology opposed those models. My explanation for this is that neuroscience was still incompatible with the demand of a robust experimental practice for psychology, which demands variables that can be manipulated. Skinner, anyway, remained against Cognitivism even in the 90's, when new techniques (PET, fMRI just to mention some) and more advanced computational models were available.

Second, notice that the concept of "function" remains the Behavioristic one, there is no return to James' conception. In fact, within Lashley's reflection the concept of "function" is applied to that which the mechanism does, but there is no reference to that which the mechanism is or the modality of its actions. Moreover, this is the only option consistent

with cybernetics as a theory of control[117].

§4. A New Model and the Formulation of Cognitivism

In the rest of the chapter, I outline briefly how the intuitions and the terms described in the previous paragraph are integrated into the other disciplines that form Cognitive Sciences. I will not develop a fully historical accounts, since there are already books that do this in far more details than those I can afford here. In line with the purpose of my dissertation, I explain some reasons for which these ideas are transmitted to other fields and in the end to Cognitivism, and why they are relevant for the computational model of mind. In this way, I start to prepare the discussion of the next two chapters, and I have the occasion to emphasize some aspects that turn out to be relevant for my criticisms to "computational representations" as explanatory tools for intensional reference.

One of the crucial points of Lashley's program was the translation of mental contents into procedural features and objects of those processes. Propositional logic and engineering of automated calculation provided a mathematical way of doing this, together with the way in which Alan Matheson Turing faced a mathematical problem regarding the theory of functions.

Charles Babbage (1791-1871) was the first able to construct an automated calculating machine: the "difference engine", finished in 1821. This is the first *computer* in the etymological sense: a machine that does mathematical calculations. This may be of interest even if trivial, since there are issues in understanding what it means to "compute" in the computational theory of mind. Even more important, Babbage was designing also another machine between 1834 and 1837: the "Analytical Engine", that was never finished due to lack of funds. The first machine was projected for executing just one kind of calculation, the second was a *general-purpose machine, i.e. a machine that can execute all the operations that match a certain set of features*. In the case of Babbage's machines, whatever it is decided to be a "calculation" in arithmetical sense. The concept of being "general-purpose" describes modern computational devices as well, thus the reader must remember this concept.

[117] Aspray (1985): «The importance of functionalism to information theory is clear. It concentrated on the functional operation of the brain, and represented the brain as a processor (of information) – as a doer as well as a reflector. Behaviorist psychology, by concentrating on behavior and not consciousness, helped to break down the distinction between the mental behavior of humans and the information processing of lower animals and machines». Given that information theory is one of the main ingredients for the computational "mental representation" and for computing devices in general (but they do not imply each other, see Piccinini and Scarantino 2010), according to Aspray the contribution of functionalism to Computationalism is that the former helped to translate mental behavior in terms of behavior of internal processes.

The interest in Babbage's machines lies in the prospects they open, which were grasped very early by Ada Lovelace (1815-1852), a noble woman who became interest in Babbage's work. Because of her interest, she translated the review of Babbage's work by Luigi Federico Menabrea (1809-1896) – Italian engineer, general and diplomat. Around 1842, Menabrea reviewed the project of Babbage's analogical machine (the Analytical Engine), and Lady Lovelace wrote some notes to the translation not only explicative, but theoretical too. In a passage from those notes, she foresaw that Babbage's machine could automatically process every element whose relations could be represented through mathematical relations and as such that they fit the structure of the machine at issue[118].

Lady Lovelace had music in mind when she stated this claim, she could not have logical calculus of propositions in mind, since it would have not come until (conventionally) 1847, when George Boole published his *The Mathematical Analysis of Logic: Being an Essay Towards a Calculus of Deductive Reasoning*, and has to wait Gottlob Frege (1848-1925)'s *Begriffsschrift* (1878) and Rudolf Carnap (1891-1970)'s *Logical Syntax of Language* (1937) for reaching an advanced stage. Nevertheless, Lady Lovelace grasped the crucial point: before whatsoever can become an argument for the automated operations of Babbage's machine, it must be made conform to mathematics, meaning that *either it must be made mathematical, or it must be expressed mathematically*.

Two points are important to be noticed here. First, the above-mentioned condition introduces *the problem of appraising the difference between "being expressed by" and "being something"*. Babbage's automated machine uses operations that *map* the mathematical relations, but those operations *are not* mathematical: they are mathematically planned and then mathematically described, but they are not mathematical unless some user interprets them as such. This is one of the things that may sound extremely trivial, but I think is worth to be stressed, because very easily – or at least, this is how it goes according to my experience – philosophical literature on Computationalism and Cognitivism discusses computing machines as "reading", "writing", "performing games", "doing computations" in the sense of "doing mathematical operations" and similar claims. Thus, at first one can have the illusion – or at least, again, this is the impression I had – that the operations are given as such (described as such) to the artificial agent that performs them, but I am going to show in *Chapter Second* and *Third* that this is not the case at all: all that

[118] See Lovelace 1843, p.694:«The operating mechanism can even be thrown into action independently of any object to operate upon (although of course no *result* could then be developed). Again, it might act upon other things besides *number*, were objects found whose mutual fundamental relations could be expressed by those of the abstract science of operations, and which should also be susceptible of adaptations to the action of the operating notation and mechanism of the engine». Italics in the text.

is ongoing in a computational machine is a self-controlled process in a specific technical sense, hence either this kind of processes has the property of representing, or it has not, *tertium non datur*, and for this point to be evaluated, the difference between "being" and "being expressed/describable by" must be kept in mind, or one ends up with claims on metaphors. My second remark is as follows. Lady Lovelace could not have in mind propositional calculus, since it has to come yet, but the idea of a logical computer was intuited before the means were available, and Lady Lovelace indicates the proper reason for which engineer of automated calculation and propositional logic are relevant for each other within the problem of a computational theory of mind.

The authors who attempted to make thinking a matter of calculation were interested in formalizing demonstrative thought. Frege helps in formulating this purpose and understanding its implications. Frege as well as Boole thought that that which counts concerning this species of thinking are logical relations between parts of propositions and propositions themselves, since the truth of propositions depends on them[119]. If this is the case, then algebraic propositional calculus is an effort of reproducing that which Frege calls «conceptual content», which is defined as the content whose changes cause a change of the truth-value of the consequences of propositions[120]. In other words, the interest lies in describing algebraically the connection between variables in a form unvarying regarding differences in the way contents are expressed, as for example emphasis for pragmatical purposes. Since connotation and variations on expression remain by definition outside every logic that aim to be formal; since that which is denoted is not considered too, because every element is considered as an undefined variable in an argument considered as an unvarying structure of relations; it follows that all that remains are variables and their connection. The line composed of certain variables and certain relations in a certain order, which is constant despite any substitution of variables, Frege calls it «function»[121].

Variables and operators that compose a function are all *signs*, hence role and meaning of signs can be individuated only on the ground of *their syntax*, *their shape* and *their past background knowledge*, unless one presupposes their interpretation. Thus, understanding and developing functions in Fregean sense *is a matter of procedure*. For example, unless one *presupposes the interpretation* of "$\rightarrow$" as meaning "the term on the left is connected to the term on the right by the relation of implication", the expression "$(\neg a \wedge b) \rightarrow c$" can be solved only procedurally, i.e. understood as an *instruction* like this: "if it is given the

[119] Cfr. Frege 1878/1967, pp.11-12.
[120] Frege 1878/1967, p.12.
[121] Frege 1878/1967, p.22.

case that you find the shape '¬a' or that which is set to correspond to '¬a' together with 'b', then pass in some sense to 'c'". There can be no other option, since signs like those have only three elements: spatial shape, spatial disposition and interpretation. This follows from their nature of being objects for a subject: either the subject connects something beyond the signs to them, or it does not. That which should be connected is interpretation, but I showed that an algebraic writing of propositions rules out exactly interpretation when it is formal, hence signs can be differentiated *exclusively* according to their sensible properties, and/or for their temporal and spatial disposition; the former corresponds to *shape*, the latter to *syntax*, as I wanted to show.

A *past background knowledge of the signs* is also needed. Shape, relation of function and content relatively to syntax must be known in advance, since these are not included in the shape of signs: they need to be individuated according to that, hence these things must be presupposed, insofar as they are the conditions for identifying and operate correctly with those signs; in fact, it is not possible to identify something as something else if that which identifies is not given. This would amount to identify something as a member of a set without specifying the features of the set. Consider the following argument too. If there was no consequence in receiving the shapes signs are, that is they would not send to anything beyond themselves neither practically nor discursively, then signs would be just objects, not actual signs, since the minimum requirement for being a sign is to send to something different from itself. Since any declarative knowledge has been excluded, the past background knowledge cannot suddenly generate a discursive content; thus, the only alternative is that receiving those signs has practical consequences. Thus, as I wanted to show, formal propositional calculus is procedural in nature, and that procedure must be part of the past background knowledge.

Anyway, the procedural nature of algebraic transcription of thinking is also a matter of history, since it was a crucial point in the debate between intuitionistic mathematics and the so-called Logicist program[122], which I discuss further when I examine the concept of information and its role as a sign within computation in contemporary sense. Moreover, both Frege and Boole were developing their theories by making explicit reference to the line of thought that conceived reasoning as calculus, promoted by authors as Hobbes or Leibniz.

If I am right, then the propositional calculus developed by Boole and Frege is the mean that allows to make thinking a procedural matter, and this is the reason for which it is

[122] About intuitionistic mathematics and Logicism, see Hilbert 1922/2007; Patton 2014.

relevant in the formation of the computational model of mind. Propositional logic and engineering of automated calculation make possible one of the core principles of Lashley's program of explaining mind in neuroscientific terms: its behaviorism consists of turning relations between contents and contents themselves into procedural features and development of processes, and propositional calculus together with a machine for automated calculation make concrete the idea of a self-organized operative being that can be described in purely mechanistic terms.

The availability of a procedural calculus anyway does not solve the problem of making such a calculation a subject suitable to be managed by machines. Here is where Turing came into play. Alan Matheson Turing (1912-1954) was seeking a solution for a purely mathematical problem proposed by David Hilbert (1862-1943), the "*Entscheidungsproblem*"[123]. Amazingly, in his work in 1936, *On Computable Numbers, with an Application to the Entscheidungsproblem*, Turing invented the first computational approach to a logical problem, together with the means to implement it, setting completely from the beginning the fundamental concepts of computer science, hence of Computationalism in philosophy of mind as well.

I will discuss Turing's work in details in the next chapter, so I mention only that the operationalization of thinking in Turing's work is grounded on fundamentals features of the theoretical, hypothetical machines named "Turing's Machine" (TM from now on) Those machines are "theoretical" because they are mentally conceived. It will become a concrete device only when the "IAS computer" will be realized at the Institute of Advanced Studies at the University of Princeton, in 1951. Its functional organization and hardware architecture, called "Von Neumann Architecture", is still the basic hardware organization in contemporary computers[124]. The fundamental features that make possible to operationalize thinking, insofar as it is possible to express it into a formal language as Boole's or Frege's, so that it can be managed by computational devices are:

1. the being "finite" (in a technical sense) of the states of the TM;

[123] In 1904, during the Second International Congress of Mathematicians held in Paris, Hilbert asked the question whether it is possible to establish for each well-formed formula (wff) of a formal language if it is affirmed or refuted given a set of premises. It is called "problem of decidability" because the point is proving for each possible wff if it is either true or false, it must be possible to decide its truth-value. Gödel's theorems are an attempt of solving this problem too (in fact, Turing refers to them in his paper).

[124] Sometimes, EDVAC is credited with being the first computer with Von Neumann architecture, but Eigenmann and Lilja 1999, p.6 states that EDVAC had not a hardware architecture fully conform to Von Neumann architecture. Eigenmann and Lilja justify this difference and the construction of IAS itself to the disagreement on the functional organization of the machine between Von Neumann and the two other engineers from the University of Pennsylvania leading the EDVAC project: John Mauchly and Presper Eckert.

2. the concept of "effective procedure";
3. the possibility of dealing with instructions as if they were data.

The procedural nature and features of propositional calculus make available its implementation by a purely mechanical device, but since this device has no access to signs in regular sense – i.e. it has no way of *internalizing signs as signs* in human sense – there was a demand for a compatible *internal language* for those devices, and *information theory has the role of providing a middle term for making meaningful – in the special sense of being operationally significant – the signals in computing machineries*. "Middle term" because by providing significance to the syntax of signals, information theory connects a purely physical stimulus with a semantic value and the function of being an instruction in a consistent sense.

The historical pattern was not directly centered on the problem of an "internal language". For the sake of historical accuracy[125], it must be considered that the pattern of the conjunction between neuroscience and informatics is as follows: first, Norbert Wiener had elaborated mathematically the concept of *feedback* and automated control in devices for military purposes in 1918, but he did not applied it to the problem of organic life until 1931; then Alan Turing conceived his theoretical device in his seminal paper in 1936; McCulloch and Pitts – which we are going to meet in *Chapter Third* – will apply Turing's work to a theoretical model of neuronal nets in another crucial paper in 1943, and they will open the problem of what it means for a neural net to compute; then, Von Neumann will concretely realize an automated calculating machine who is equivalent to a general-purpose Turing Machine with his EDVAC (1946), or IAS (1951), depending on which historical thesis one embraces; Shannon elaborated and published his theory of information in a groundbreaking paper in 1948. All of those will make available the means for conceiving living beings in terms of self-controlled devices, whose internal operations are *communications* between areas, whose difference consists of being devoted to different functions. This last passage will come only with Norbert Wiener's synthesis in his central book *Cybernetics, Or the Control and Communication in the Animal and the Machine*, in 1961 (second edition, the first was published in 1931), but «the first person to use information theory to describe *the mind as a whole* was Donald Broadbent. But George Miller was the first to apply it within (a more limited area of) psychology»[126] in 1958 and

[125] The following references are just a summary of Boden 2006's reconstruction of the passage from Turing's work to Wiener's synthesis. For the extensive enquiry on the history of this passage, see Boden 2006, ch.4, pp.168-237. For more on the contribution of information theory to engineering of automated computation, see also Aspray 1985.
[126] Boden 2006, p.283.

1956 respectively.

Thus, it is Wiener who grasped the intersection between computing and information theory within the problem of a theoretical neuroscience/biology, and he did it within the problem of *control*, without any direct reference to information as an "internal language": his problem was how different parts could communicate in order to coordinate their activities and perform their functions for a complex end to be achieved. Under this respect, Shannon's information theory constitutes a theoretical, stochastic mean for describing in terms of internal flow of signals that which happens within an environment through a specific strategy, which I discuss extensively in *Chapter Second*. For the purpose of the present enquiry, however, it is preferable to begin since now to discuss information theory as the internal language of computing machines, because this is that which concretely is for those devices, as I am going to show in proper details in the next chapter.

I expand the topic of how information theory joined the other disciplines we have been surveying. As I said, engineering of automated calculation, mathematics, propositional logic and neuroscience went together quite naturally (logically) into McCulloch and Pitts' work[127]. Despite this, according to Margaret Boden, it was not McCulloch and Pitts' work per se who gained attention by biologists and experimental sciences of mind in general[128], even if at least McCulloch was interested in biology[129]. McCulloch attended the seminars by Clark Leonard Hull (1884-1952), who attempted to explain Tolman's result in behavioral terms[130], and specifically, Hull deepened Lashley's idea that the explanatory model stimulus-response can be posited *within* the organism, instead of as external behavior. According to Boden, Hull was the most famous Behaviorist of his days, but she also points out that Hull's exposition limited the understanding and the spreading of his theories, since he relied on a formal language and on a theoretical approach that were not widely shared by his colleagues[131]. This is the reason for which I defined Behaviorism through Watson's and Skinner's work, without mentioning Hull's research.

McCulloch and Pitts' work was not immediately acknowledged as the revolutionary idea it is for multiple reasons. Just to mention a couple of reasons, the influence of Behaviorism was still at work, and in their paper the authors claimed that their system could provide all

[127] See also Boden 2006, p.190.
[128] See Boden 2006, pp.194-195: «what eventually brought the paper to the attention of psychologists capable of appreciating it was its role in the design of the modern computer […]. As McCulloch himself admitted, ''as far as biology is concerned, it might have remained unknown'' had it not been picked up by von Neumann and used to design computers».
[129] See Boden 2006, p.195.
[130] Boden 2006, p.261.
[131] Cfr. Boden 2006, pp.261-262.

the answers to psychological questions; since they were discussing internal processes as well as internal contents, such a statement was hard to accept for the community of psychologists at that time. Moreover, there were also issues with the formal notation they used: they adopted a revised version of Rudolf Carnap's formal language, but many commenters agree that it was uselessly complicate and that some passages included mistakes[132].

That which made the fortune of McCulloch and Pitts' paper was their capacity to *embody* logical operators into circuits, and the parsimony of the neuronal model proposed, i.e. the fact that McCulloch and Pitts' theoretical artificial neurons have two values, as real neurons: either they discharge or not, and this can be translated in the simple binary code included in Shannon's information theory[133]. Surprisingly enough, the authors were clueless on all of those when they wrote their paper: they were not acquainted with Shannon's theory, nor with engineering in general, nor with the problem of automated control, nor they could have in mind concrete TMs, since there were none according to public knowledge (their construction was developed in military laboratories). Thus, it was Von Neumann who understood the conjunction between McCulloch and Pitts' proto-theoretical, computational neuroscience and Shannon's information theory[134]; whereas it was Norbert Wiener who systematized a unified framework for the five disciplines who came together to form the early nucleus of Cognitivism.

The Hixon Symposium was the occasion in which most of those people were able to meet and notice that they were sharing common interests as well as common theoretical tools and frameworks[135]. Successively, they were Von Neumann and Wiener who organized a meeting at Princeton about the arguments that would have later formed the cybernetics, and it was Wiener who organized a further conference through the Macy Foundation about the problem of feedback, respectively in 1944 and 1946[136]. The most important event for history of Cognitive Science, anyway, is considered to be the Dartmoor Symposium of Information Theory promoted by the MIT in September 1956, in which crucial papers were presented at the public, which definitively challenged and refuted traditional Behaviorism and presented new ways of approaching problems in different fields with identical

[132] Fitch (1944), Piccini (2004), Schlatter e Aizawa (2008).
[133] Boden 2006, pp.195-196.
[134] For more on the joining of neuroscience and Shannon's information theory, see Dimitrov, Lazar and Victor 2011.
[135] Gardner 1985, pp.23-24.
[136] Gardner 1985, pp.24-27.

theoretical tools[137]. These and other meetings contributed to create a self-aware community of people who were working on a specific direction, different from that adopted until those years.

Those events fostered an interdisciplinary discussion, which in turn became the basis for the syntheses proposed by George Miller Eugene Galanter and Karl Pribram, with their collective work *Plans and the Structure of Behavior* (1960), or by Wiener with the second edition of his *Cybernetics*[138], and in the end the book *Cognitive Psychology* by Ulric Neisser (1928-2012) in 1967[139]. Cognitive science raised from this interdisciplinary work, and as I briefly sketched it is a work that comes from neuroscience, engineering, logic and mathematics and only *in a second moment* it becomes an accepted paradigm in psychology, sociology and other sciences[140]. There is a certain order in this: Cognitivism raises as an opposition to Behaviorism, even if it is prepared by Behaviorist neuroscientists, and it comes back into psychology after traveling within the fields we surveyed.

§5. Conclusions of Chapter First

In this chapter, I discussed three topics. First, the relation between Behaviorism and Cognitivism. I showed that the main motive of Behaviorism is an epistemological concern: the possibility of providing a robust experimental practice for psychology, and the consequent rejection of mental features and events on the ground that they are not variables suitable to be manipulated during experimental checks of theories. The rejection of mind on this ground can be expressed by three questions: whether mind is observable; whether mind as the object of psychological enquiry allows to individuate and manipulate the causally relevant variables, so that a proper scientific, experimental practice can be provided to psychology; whether it is necessary to admit a causal efficacy of mental states.

The second topic I discussed is the elaboration of a new model of mechanism within neuroscience and the crossing of several other disciplines. The Cognitivist answer to the questions of Behaviorism comes from the new model I outlined in §3 of this chapter. It provides a positive answer in a twofold way: it makes mind observable *by moving the focus on the internal processes that realize events with the same function and role of mental events*; it *provides mathematical, sharp theoretical framework and tools for defining its concepts, objects and theorems*, so that an experimental practice on mind can be managed.

[137] Gardner 1985, pp.28-29.
[138] According to Boden, there were several editorial problems with the first edition, so even if Wiener's theory was known and respected, its proper formulation should be sought in his revision of his seminal work.
[139] Gardner 1985, pp.32-35.
[140] Boden 2006, pp.282-283.

In other words, the cybernetic synthesis started in neuroscience and biology that which Newton did for physics: it made mathematical and rigorous the study of cerebral structure and cerebral activity; moreover, it did this through an unparalleled interdisciplinary cooperation. I say that the contribute was to biology and neuroscience because I showed that the new model of mechanism was born as a research program for a theoretical approach to life, centered on a new understanding of the concept "behavior" (as resulting from internal control of responses), to which the raising computer science – which included automaton theory and research on artificial intelligence – marched side by side, even if as a new field of enquiry on its own. The nature of the new model was closer to neuroscience and biology than to psychology, to which this program arrived in a successive moment.

About the being close to biology and neuroscience of the new model, it is interesting to notice that it stems from researches in Behavioristic neuroscience and attempts of applying computer science and information theory to neuroscience (proto-computational neuroscience), and only successively it came back to psychology, through the works of George Miller, Donald Broadbent, Noam Chomsky and others. I showed that Pavlov widened theories prepared by his teacher Sechenov, and he was the first to posit the topic of behavior and its conditioning. Behavioristic psychology receives his lesson in terms of overt behavior and its conditioning, and uses the ideas from neuroscience to overcome functional psychology with a subversive strategy, that is by relying on the reference of Functional psychology to biology through two Darwinian concepts: "adaptation" and "natural selection". The latter concept allows to conceive the role of mental events and activities according to a relation with environment and contingent situations; the former has different purposes in different theories. In Functional psychology, "adaptive action" is still faithful to its original use, but Watson expands its meaning to include individual acts whose purpose is to allow individuals to fit into contingent situations. Lashley did not speak of adaptive acts, but while he and Hull move the problem of behavior from overt acts to internal organization, at the same time integrate the idea of adaptation in Watson's sense into neuroscience in terms of internal control of responses to stimuli.

In the end, notice that the key concepts of the new model concern internal structures, their function and organizations, no one was talking about experiential or phenomenal features of mind, and no one will do: the interest was in describing the *efficient causality of biological structures for implementing functions, features and results of mental events and activities*, namely *the internal behavior of a mechanism*. I will show in the next two chapters that this is that which the computational representation still does. Thus, Hull's

idea of reformulating the problem of behavior in terms of internal events; Lashley's idea of understanding behavior in terms of internal organization of the living being; Tolman's idea of internal maps that shape behavior by organizing in an informative picture stimuli from environment, those are the core ideas that make possible the Cognitivist synthesis across multiple fields, since they provide a basis on which their relevance for each other can be conceived.

In fact – and this is the third and last topic I discussed – the new model is grounded on specific features that are not present in previous works, especially those of the mechanistic tradition in physics. From this point of view, biology and physiology offered an important contribute to our understanding of the world, and mathematics offered tools for its formulation that cannot be replaced. The features in question are the following:

1. *a different model of causality*: causality in classical mechanistic models was monotone, that is it proceeds in a progressive order with only one direction, namely the series of time. The new biological processes showed interaction between results realized by previous stages of the process and these same stages, which get in turn modified by them. Stated otherwise, the new model of causality includes processes that proceed by taking into account their own results and modify their development accordingly. This made viable thinking purposeful behavior in natural terms[141];
2. *hierarchical organization, cooperative interaction, integration*: the authors I mentioned contributed to the cybernetic synthesis with many ideas. The new mechanism is composed by areas dedicated to specific functions, which cooperate for realizing the more complex activities and are able to integrate the results from other areas into their own activities;
3. *activities are planned*: this is one important point for the general purpose of my enquiry. I mentioned Tolman's idea of internal mapping and active construction of organization of distal stimuli received from environment – an idea that Tolman elaborated from Gestalt psychology. That which is said "to represent" in contemporary theories is the organization of internal structures that map distal stimuli[142] and the symbolic descriptions that serial computational devices manipulate and give as outputs[143]. Thus, it is confirmed that it is necessary to

[141] This is an important point in Rosenblueth, Wiener and Bigelow 1943. See also Perlman 2016 for a review of modern theories that relate functionalism and teleology.
[142] See for example Marr 1979/1982; Hinton, McClelland and Rumelhart 1987; Bechtel 2008, especially chapter 5; Kandel et al. 2013; Cao 2019, particularly clear are pp.284-285; Egan 2019.
[143] See again Ramsey 2006 and Egan 2019, plus Fodor 1975; Haugeland 1989; Floridi 2004, especially the section on semantic information.

understand the computational sense of "to represent" through computer science, by enquiring how computational devices work, how they implement their functions and how the processes ongoing within them are structured, as I said in *Introduction*, §1. This is true since (A) this is where computational representations come from and (B) this is where they belong still nowadays.

Chapter Second. The Classical Model of Computation

In his *The Irrelevance of Turing Machines to AI*, Aaron Sloman argues that both Computationalists and their opponents believe that «our everyday notion of "computation", as used to refer to what computers do, is inherently linked to or derived from the idea of a Turing Machine, or a collection of mathematically equivalent concepts (e.g. the concepts of a recursive function, or the concept of a logical system)»[144]. According to Sloman, this is not the case. His argument goes as follows.

Sloman stresses the difference between two traditions: one concerns the construction of *mechanisms for control*, the other of *mechanisms for performing abstract operations*[145]. In fact, in the previous chapter I emphasized their connection within the genesis of Cognitive Science, but as Sloman – and as Boden 2006 extensively documents – I do not claim that the studies in engineering about self-controlling devices and the engineering of automated calculating machines are one and the same; this is the reason for which historiography talks about a cybernetic *synthesis*. Sloman explains that machines for control were designed to do physical tasks, whereas machines for performing abstract operations were designed for performing abstract operations on abstract entities[146], hence two aspects can be distinguished in machines that include both kinds of operations: autonomy of control, that is capacity of implementing physical processes by themselves; and autonomy of information, that is autonomy in determining what to do next[147].

It is worth mentioning that Sloman stresses an important feature of their relation: «if the physical design constrains behavior to conform to certain limits then there is no need for control signals to be derived in such a way as to ensure conformity, for example. […] it is not a discovery unique to so-called situated AI, but a well-known general principle of engineering […]»[148]. This is relevant in my forthcoming discussion of how computers can "understand" different instructions and execute new programs that they have never seen before: the production of control signals through structures that determine necessarily a proper behavior is a key feature of modern computers. Sloman's observation also helps in understanding a «trade-off» that relates the two aspects differentiated above.

Sloman's argument goes on with an exam from the historical standpoint of this trade-off

[144] Sloman 2002, p.88.
[145] Sloman 2002, p.89.
[146] Sloman 2002, p.90.
[147] Sloman 2002, pp.90-91.
[148] Sloman 2002p, p.92-93.

that reveals similarities, but also different needs for facing specific problems, amongst which some relate exclusively to how implementing physical actions, other exclusively to how manage information (in a common sense, not Shannon's), and they provide further basis for classifying more or less automated machines[149]. The intersection of those aspects may raise interesting theoretical questions, that can be also relevant for building specific machines, but nevertheless the problems in *finding the actual implementation of specific physical operations* concern *equally specific* modeling, whereas Turing Machines provide a (one of many) framework for enquiring *general* theoretical questions, which is not the case of research on artificial intelligence[150]. Moreover, Turing Machines have unreal physical features, that are possible exactly for the theoretical nature of TMs, and translating the possibilities of TMs into concrete computational devices demands solutions to problems that are not included in Turing's theory, nor can be deduced from it. To be more specific, Turing's theory is relevant for studying mathematically unbounded competences and operations, but is not relevant for the real, bounded competences that animals display[151]. On the contrary, according to Sloman, concrete computers have features that make them relevant for several fields, but none of those features is connected to TMs[152].

Sloman's thesis is correct: TMs are not superimposable to concrete devices, and their mathematical and "physical" features are unrealistic, idealized; and they do not tell how implementing concretely a computational automaton. Another author, Richard Samuels, remarks that some commenters have given too much importance to Turing's contribution, and argues that probably some misunderstanding is the basis for this mistake, namely the one noticed by Fodor: probably some authors think that classic computational models of cognition are committed to claim that mind is a Turing Machine in some sense and mental activities are computable in the sense of being computed by a Turing Machine[153]. Even if classical computational models *in cognitive science* (not only in philosophy of mind) are conceived as implementing symbolic manipulation, that is abstract operations involving some kind of signs, «classic computational models need not – and should not – attribute all the properties of Turing machines to cognitive systems»[154], since TMs have infinite memory, serial procedures and deterministic state transition, that is the stage at a certain time determines completely and necessarily the state at the successive moment. Kenneth

[149] Sloman 2002, pp.92-99.
[150] Sloman 2002, p.100.
[151] Sloman 2002, pp.101-102.
[152] Sloman 2002, pp.102-109.
[153] Samuels 2019, p.105.
[154] Samuels 2019, pp.103-105.

Aizawa also makes a similar point: «the recurrent allusions to Turing machines and Turing-equivalent computing devices suggest that the history of the computational theory of mind is the history of the birth of Turing machines and their subsequent influence. Such allusions, however, are at times misleading. For one thing, there are important episodes in the history of "the computational theory of mind" where Turing-equivalent computational formalisms had no role. There are cases where the application of the formalism of Turing-equivalent computation to one or another scientific or philosophical issues was not the goal»[155].

I share all the points those authors make, but despite this I am going to open the new chapter with a detailed analysis of Turing's work, even if I want to discuss the plausibility of artificial, computational models of intellectual reasoning. I am aware and I acknowledge that "cognition is a computational process that involves symbol manipulation" is a different claim from "cognition is a computational process in Turing's sense of being computable"; as I am aware that Computationalism within Cognitivism was born from the cooperation of different fields mutually non-reducible, and that the history of the idea of control is not the history of computer science as a whole, or of mathematics, or of Turing's idea.

Nevertheless, Ramsey's remark about the necessity of being informative of the concepts applied leaves room for making meaningful the choice of starting the enquiry on computational models from Turing's work today as well as in the 60's of the previous century. For if "computational" must mean something for a theory of cognition, it must be defined according to hypothetical or concrete computation as it is described and discussed in computer science, or it becomes some new concept with unclear meaning, as "representation" is threatened to be. Thus, Turing's work is relevant because, as I am going to show, concrete computers and computational models presupposes the theory developed by Turing under crucial respects, not last the definition of the fundamental concepts of the field, and Turing's theory has precise constraints. Knowledge of these relations is also important for appreciating the difference with the connectionist models of computation[156], and for answering the question of computation as symbolic manipulation.

That which I have just said *does not* mean that all the features and all the constraints of TMs are features and constraints of modern computers and modern artificial computing networks; it just means that some features and some constraints of the theory of TMs are still the basis on which computation is described, understood and studied in computer

[155] Aizawa 2019, p.65.
[156] See the passages on what it means to compute for neural nets in *Chapter Third*.

science, hence it must be taken into account in any theoretical discussion of "computational models of cognition". Since my discussion *is* theoretical, because it concerns what representing in a computational sense means and in turn how computational devices work and what their principles are, discussing Turing's work is relevant for understanding fundamental principles, their consequences and their context, but one must be aware of the important remarks made by Sloman, Samuels and Aizawa.

I must also ask the reader to be patient, since it cannot be clear at a first glance what the significance of the exposition in §1 is, but once that he/she reaches §2, he/she will see that the theory of TMs is identical to the functional, concrete organization of computers, hence that which is exposed as a formal theory here will provide a deeper understanding of the principles that lie the foundation for (the classical) concrete devices.

§1. Alan Turing's Theoretical Automated Computer

1.1. Theory of Non-General-Purpose Turing Machines

Let us begin by defining the *components* and the *basic functions* of the hypothetical Turing Machine. Notice that there is a slight difference between *On Computable Numbers, with an Application to the Enstscheidungsproblem* (1936) and *Computing Machinery and Intelligence* (1950). The two ways are equivalent, but their organization is different regarding how the functions they implement are distributed. Here components and basic features are listed, and the above-mentioned difference is indicated:

- TMs have a *tape*: the tape is a physical support for writing and reading, thus it is both the source of input, the support for outputs, and the *working memory*, the memory on which intermediate passages for carrying on instructions are registered, where this is required[157]. The tape is supposed to develop horizontally and to be divided in discrete sections[158]. In each section one and only one symbol is

[157] Tape in TM's is a "general purpose" memory: it is equivalent to all kinds of memory in modern computers, which are instead differentiated according to their function, and this bears differences in architecture between the various memories, but nevertheless the principle of their role is exactly the same. When the topic of memory will be discussed in the following paragraphs, the reader must have in mind this difference.

[158] In Turing 1950/2004, p.444 it is said: «we have mentioned that the "book of rules" supplied to the computer is replaced in the machine by a part of the store. It is then called the "table of instructions". It is the duty of the control to see that these instructions are obeyed correctly and in the right order. The control is so constructed that this necessarily happens». Thus, that which is added compared to *On Computable Numbers...*, is that the set of instructions is written on the tape, hence it is a dedicated part of the memory of machine at the beginning of its operations. Even if – as Von Neumann himself declared – it is the hypothesis of Universal TMs that inspired the hardware organization of EDVAC and not the opposite way round, it is plausible that Turing introduced this change in consequence of the publication of technical schemas describing EDVAC. In fact, the strategy used by Von Neumann to make EDVAC universal is exactly treating instructions as data stored in the internal memory.

written[159]. Robič 2015 explains that many variants of the tape[160] are possible, but he also shows that all of them can be implemented in a standard TM[161], thus I will discuss only this model. In Turing 1936 *there are symbols* written on the tape, whereas in Turing 1950 it is said «the store is a store of *information*, and corresponds to the human computer's paper (my note: TMs are idealized human who compute, namely human *computers*), whether this is the paper on which he does his calculations or that on which his book of rules is printed»[162]. This change also supports the thesis that some difference in how TMs are described depends on the publication of technical schemas of EDVAC;

- TMs are capable of "scanning" one section of the tape for each time, in which there is just one written content (symbol), or as it is said in Turing 1950/2004, TMs have a "*executive unit*": in Turing 1936, Turing only says that the part of the tape "in the machine" is a "scanned section"[163], whereas in Turing 1950/2004 it is said: «the executive unit is the part which carries out the various individual operations involved in a calculation. What these individual operations are will vary from machine to machine»[164]. The difference between the two versions is that in Turing 1936 the TM is considered as a "black box" – a part of the tape is into the relevant component, and something unspecified execute the relevant operation on the tape, somehow – whereas in Turing 1950/2004, we find a dedicated part for executing the operations, and this is more conform to the actual architecture of concrete computers, as it is the fact that instructions are given as data. The "black box view" of TMs does not affect in any way the mathematical theorems that this theory allows to deduce[165], but one can see from this that Sloman has his reasons for his remarks. I will tell more on how TMs operate and what their operations are below;
- Only in Turing 1950/2004, TMs have also a *control*[166]: this is close to modern CPU (Central Processing Unit), but not exactly equivalent, since modern CPUs also retrieve instructions, whereas Turing originally discussed only TMs that execute the series of instructions provided by an operator.

[159] Turing 1936, p.231.
[160] Robič 2015, pp.107-108. It is possible to posit finite tapes, multiple tapes, multidimensional (bidimensional) tapes.
[161] Robič 2015, pp.109-113.
[162] Turing 1950/2004, p.444.
[163] Turing 1936, p.231.
[164] Turing 1950/2004, p.444.
[165] Cfr. Minsky 1967, p.13.
[166] See again Turing 1950/2004, p.444.

One of the central features of TMs is the being *finite* of its operations, arguments and states, *both in time and space*. The tape is divided into discrete sections which are indexed by integer, natural numbers. Time is conceived according to the same principle. This abstraction allows to ascribe each stage of the process to one and only one time $t \in \mathbb{N}$. Conventionally, t_0 is the time at which the device is set in motion. Then, a further condition is imposed: that the machine has a *finite number of states*, that is, each "situation" in which it may be at a given time *t* belongs to a set with a limited number of elements[167]. Turing calls "*m*-configuration" or "conditions" each member of this set $Q = \{q_1, \dots, q_m\}$ defined in $\mathbb{N}$. They must be limited in number and discrete (as opposed to "continuous"), otherwise they could not serve their purpose, i.e. being unambiguous instructions for equally specified situations[168]. Analogously, the set of possible symbols $S = \{s_1, \dots, s_n\}$, defined in $\mathbb{N}$ too, must be also finite: if they were infinite, they would become increasingly similar and the more their number increase, the more and more likely is that one or more of them become indistinguishable[169].

Wilfrid Sieg helps in giving a precise definition of the main features of how TMs operate. The first two points paraphrase the previous explanation, the third expresses the most important feature regarding how TMs operate, the last two points add further constraints that Turing also indicated. The main features of how TMs operate are[170]:

- (**Boundedness.1**) «there is a fixed bound on the number of symbolic configurations a computer can immediately recognize»[171];
- (**Boundedness.2**) «there is a fixed bound on the number of internal states a computer can be in»;
- (**Determinacy**) «a computor internal state together with the observed configuration fixes uniquely the next computation step and the next internal state [*m*-configuration]». This condition is called "determinacy" because it prescribes that operations of TMs are *necessarily* fixed by the two elements indicated, a pair composed by a symbol and a *m*-configuration;
- (**Locality.1**) «a computor can change only elements of an observed symbolic configuration»;

[167] Turing 1936, p.231.
[168] Turing 1936, pp.250-251.
[169] Turing 1936, p.250.
[170] Sieg 1998/2016, pp.396-397.
[171] Sieg 2018, p.392 points out that this limitation was introduced by Turing for psychological reasons. Cfr. in fact, Turing 1936, pp.250-251.

- (**Locality.2**) «a computor can shift attention from one symbolic configuration to another one, but the new observed configuration must be within a bounded distance from the immediately previously observed configuration»[172].

All that I have been explaining so far serves the same purpose: making univocally describable procedure, its argument and its result for each time *t*, from t_0 to t_n. It is preferable to fix the meaning of "*m*-configuration" before starting the technical part of the paragraph, through which I explain the importance of the univocity achieved through the previous constraints. Each "*m*-configuration" is an equivalent to an *instruction stored in a part of the memory* in modern computers. This is the argument for this interpretation:

- When the term is introduced, it is said: «we may compare a man in the process of computing a real number to a machine which is only capable of a finite number of conditions $q_1, q_2 \ldots, q_m$; which will be called "*m*-configurations"»[173];
- Turing offers three arguments for demonstrating the extension of "computable numbers"[174]; the first is "intuitive", the third the most rigorous. In the first exposition Turing writes that «the behaviour of the computer[175] at any moment is determined by the symbols which he is observing, and his ''state of mind'' at that moment». The same function is ascribed to "*m*-configuration" at the beginning of the article, in the line just quoted. Moreover, it is explicitly stated that «to each state of mind of the computer corresponds an ''*m*-configuration'' of the machine»[176];
- In the third exposition, it is said: «but we avoid introducing the ''state of mind'' by considering a more physical and definite counterpart of it. It is always possible for the computer to break off from his work, to go away and forget all about it, and later to come back and go on with it. If he does this, he must leave a note of instructions (written in some standard form) explaining how the work is to be continued. This note is the counterpart of the ''state of mind''»[177];
- thus, "*m*-configurations" is not a state, as at the beginning of the paper, or in the first exposition concerning the extension of "computable numbers". "State" and similar expressions occur only where an informal way of speaking occurs. In fact,

[172] Turing 1936, p.250.
[173] Turing 1936, p.231.
[174] For the purpose of this section, "computable numbers" can be intuitively understood as "numbers that can be listed by listing the values of a function executed by a Turing Machine".
[175] In this occurrence, "computer" means "a person who computes". When Turing wrote this paper, equips of mathematicians were doing computations with paper and pencil. See Minsky 1967, p.108 for the same interpretation.
[176] Turing 1936, p.251.
[177] Turing 1936, p.253.

the third exposition, which Turing regards as the most rigorous, uses as synonymous of "*m*-configuration" an *instruction, to the extent it is the content* (in some sense) *of the machine at a certain moment*[178]. Thus, "*m*-configuration" must be understood as a *disposition to execute an instruction*, more than as a "state" in general. It is worth noticing that in concrete devices it is better to talk of "states" when the reference is to concrete situations and of "instructions" when the reference is to functional descriptions. More on this will follow.

Once that the previous constraints are fixed, the operations of a TM at a certain time t_n can be described by the function that associates the pair symbol s_n and operation q_n, to a response R in t_{n+1}, namely $R(t+1) = F(q_t, s_t)$[179]. The pair "q_n, s_n" is called "*configuration*" of the machine[180]. This function describes the *response* of the TM to a given *configuration*, and it states that the response at a given time *t* is determined by the scanned symbol and the *m*-configuration in *t-1*. It cannot be otherwise, since the constraint «the possible behavior of the machine at any moment is determined by the *m*-configuration q_n and the scanned symbol *S*(*r*)»[181] holds for all *t*. For the same reason, given that which it has been said about *m*-configurations, it follows that it is possible to define a function *G* such that the *m*-configuration in *t+1* derives from the configuration in *t*. Thus, we have: $Q(t+1) = G(q_t, s_t)$. Now the reader can appreciate that the constraints of finiteness are necessary to allow these basic principles being necessary. Functions *F* and *G* describes the state transitions of a TM. When he refers to these functions concerning finite-state automata, Minsky calls them «"transition" functions»[182]. Since they are the same as at p.118, where the author discusses Turing Machines, I will refer to them as "transition functions".

For the transition described by the transition functions to be fully described, it is necessary to relate them to a further set: the finite set D of the operations that a TM can execute. In fact, the response *R* of a TM consists of both a symbol *s*, a new state *q*, and an action *d*. According to Turing, the operations are only: (d_1) reading, (d_2) writing into a blank square, (d_3) moving the tape right or left (given the initial presupposition of a fully horizontal tape), (d_4) changing a symbol into a tape that is not blank; (d_5) halting under certain conditions.

[178] See also Priestley 2011, p.79: «The *m*-configurations represent the computer's knowledge of which steps in the computation have been performed and what is to be done next, but not the results of those steps».
[179] Minsky 1967, pp.117-119. I am following Minsky 1967 in defining this function as well as in the following explanations, since I found his notation smoother that Turing's original one, and his explanation of the Turing's machine easier to follow.
[180] Turing 1936, p.231.
[181] Turing 1936, p.231.
[182] Minsky 1967, p.20.

Thus, there is the set $D = \{d_1, \dots, d_5\}$. Turing points out that «it is my contention that those operations include all those which are used in the computation of a number»[183]. For the constraint "Determinacy", that holds for the other functions too, it is possible to define the function $D(t+1) = D(q_t, s_t)$[184]. Since the action is set by the configuration, as any other component of the response, *D must be* finite. There are as many possible responses as the combinatorial calculus of *m*-configurations and symbols allows; thus, if the sets *Q* and *S* are finite, then $Q \times S \rightarrow D$ [185] can produce just a finite number of outcomes $\delta \in \mathrm{D}$.

Given the previous definitions and the constraint of "Determinacy", the reader can notice that the behavior of a TM at each time *t* of every TM can be described as a line as: $(q_i, s_j, G(q_i, s_j), F(q_i, s_j), D(q_i, S_j))$[186]. In fact, all that the machine does depends on its configuration at a certain time, and all it does is scanning a certain symbol while being in a certain *m*-configuration, which in turn causes it to do one of the actions defined in *D* and to move into another configuration and scan the next symbol.

The line described above is nothing over and above the formal description of that which I just said according to the notation stipulated by Minsky. The quintuplet in question can be used to define a function φ that describes the behavior of the TM using all the elements defined during this section; thus, we have: $(q_k, s_k, d_k) = \varphi(q_i, s_j) = \big(G(q_i, s_j), F(q_i, s_j), D(q_i, s_j)\big) = q_{ij}, s_{ij}, d_{ij}$. The code "*i, j, k*" in subscript translates the succession of time as a relation between previous and successive states. This translation summarizes the idea that the behavior of the machine depends on the association between states and symbols at each discrete time. I hope the translation will help the reader in understanding the affinity between TMs and concrete computers, when I discuss the fetching-decoding-executing cycle.

1.2. Preparatory Definitions to the Theory of Universal Turing Machines

I am going to introduce some definitions, each on its own. Their relation concerns how to construct a *Universal Turing Machine* (UTM), which I discuss in §1.3, which is a TM able to compute every computable function that can be computed by a TM; or a TM able to

[183] Turing 1936, p.232.
[184] See Minsky 1967, p.118. The function is defined in Minsky 1967, the argument is mine.
[185] In this case, the symbol "X" between the letters that designate the two sets is not an arithmetical product, but the sign of the *cartesian product,* which is the combinatorial calculus of all the elements that belong to a set with each elements of another set.
[186] See Minsky 1967, p.119. It must be read: *m*-configuration in *t*, symbol scanned in *t+1*, *m*-configuration in *t+2* as it follows from the configuration in *t* and the symbol in *t+1*, symbol in *t+2* as it follows from the configuration in *t* and the symbol in *t+1*, motion of the tape in *t+2* as it follows from the configuration in *t* and the symbol in *t+1*. This form correspond to that which Turing calls the "*standard form*" (Turing 1936, p.239) of machine tables that report the complete configuration. For these terms, see immediately below.

imitate the behavior of every possible TM, or a TM that can compute *all functions* that describes the behavior of any TM. These are all different ways of saying the same thing. In this paragraph, I discuss also what "to compute" means in the context of the theory of Turing Machines, so that some mistakes about this can be ruled out as soon as possible.

The first point is the definition of "*standard description*" (S.D.) and "*description number*" (D.N.). Given the results q_k, s_k, d_k of the formula φ as it is introduced above, suppose to write them in a row, then let us replace each part according to the following criteria:

- $\forall x, 1 \leq x \leq m: q_x = H\ followed\ by\ x \times A$. This time the "X" means "repeat as much time as "*x*";
- $\forall x, 1 \leq x \leq n: s_x = H\ followed\ by\ x \times C$;
- Turing does not introduce any substitution for the moves, since he does not define the set D. Nevertheless, this does not change anything for our purpose, thus I can posit one like this: $\forall x, 1 \leq x \leq z: d_x = H\ followed\ by\ x \times W$;
- Turing numbers the lines of the tables, which is important, thus let us also introduce the symbols $N_{1...}N_k$ for this purpose.

The replacement will provide a series of letters of the following form: $(N_1)DA \dots A_x; DC \dots C_x; HW \dots W_x; (N_2)DA \dots A_{x\prime}; DC \dots C_{x\prime}; HW \dots W_{x\prime};$ ... and so on. This series is a *standard description* of a complete configuration. It is just a way for writing horizontally every step of the operating of the TM at stake. If successively letters are replaced with digits, then a *description number* is defined[187]. A further notation can now be introduced: given a certain TM called "***M***" and its D.N. indicated by "α", each TM can be designated by the function "***M***(α)".

There is a second definition to be discussed: the central definition of "*computation*". About this, Turing begins by giving the following definition: «if the machine is supplied with a blank tape and set in motion, starting from the correct initial *m*-configuration, the subsequence of the symbols printed by it which are of the first kind will be called the *sequence computed by the machine*. The real number whose expression as a binary decimal is obtained by prefacing this sequence by a decimal point is called the *number computed by the machine*»[188]. Later, he adds the following:

- «a sequence is said to be computable if it can be computed by a circle-free machine. A number is computable if it differs by an integer from the number computed by a

[187] Cfr. Turing 1936, p.240.
[188] Turing 1936, p.232. Italics in the text.

circle-free machine»[189];

- «a computable sequence γ is determined by a description of a machine which computes γ»[190].

For those lines to be proper understood, some history is necessary. Before Turing, "*to compute*" meant simply "to write the values of a function according to (intuitive notion of) a mechanical procedure"[191], and a "function" may be defined in two ways:

- «a *function* is a rule whereby, given a number (called the *argument*) one is told how to compute another number (called the *value* of the function for that argument)»[192];
- «a function is a set of ordered pairs <*x*, *y*> such that there are no two pairs with the same first number, but for each *x*, there is always one pair with that *x* as its first number»[193].

There is no extensional difference between those definitions, but there is a relevant conceptual one: whereas the first is descriptive and specify *what a function does by providing how the rule operate*; the second is associative and specify only *what a function achieves, but not how it does it*. The second definition conceives functions as associations between sets (or between the same set doubled), the first definition conceives functions as *procedures*, and this difference is relevant for my purpose, since Turing Machines concern procedures[194].

Turing's proposal consisted of providing some rigorous model to the *descriptive conception* of function and this model is supposed to be the one of an "actual" machinery. It can be shown by seeking the answers to a twofold question: what a mechanical procedure is, and *what kind* of mechanical procedures are suitable for defining computation. Clarifying these points is the reason for which some history is necessary. Regarding what is a mechanical procedure, there is a twofold line of research: on the one hand, there is a tradition of research on automated (executed by automaton) procedures for calculating I partly surveyed in §4 of the previous chapter, which in turn presupposes – on the other

[189] Turing 1936, p.233.
[190] Turing 1936, p.239.
[191] Cfr. Soare 1996. Soare reports from Sieg that a "computor" was some idealized human being who does calculations with pen and paper according to some "*function produced by a mechanical procedure*" (Soare 1996, p.291). Algorithms as specified procedure for hand-made calculations was regarded as "mechanical procedures" since ancient times (Soare 1996, p.288). What Turing accomplished was also to define an intuitive yet rigorous notion of "mechanical procedure" for computing (Soare 1996, p.294). Thus, according to Soare's reconstruction, such a definition was missing. The idea of algorithm was already there, but it was not rigorous yet. See also below.
[192] The difference is stated in Minsky 1967, p.133. Italics in the text.
[193] Minsky 1967, p.133.
[194] Minsky 1967, pp.133-134.

hand – the most general tradition of inventing *ordered procedures* for calculation, generically called "algorithms" in honor of the Arabian mathematician Al-Khowarizmi, and known since the times of ancient Greeks[195]. Regarding *what kind* of procedures, the history of the concept "recursion" is central for understanding the outcome of Turing's work for the clarification of "computation". Starting from Dedekind work in 1888, *The Nature and Meaning of Number*, Kurt Gödel (1906-1978) defined the formal schema of general recursive functions in 1934, on the basis of his work in 1931 and the suggestions of Jacques Herbrand (1908-1931). Later in 1936, Alonzo Church (1903-1995) published his famous paper *An Unsolvable Problem of Elementary Number Theory*. Here Church introduces the notion of "*effective calculability*", in the informal sense of being actually calculable through some algorithm, as a synonymous of his being λ-calculable (that is, calculable within a formal symbolic calculus he developed) and of Gödel's computability by general recursion[196], thus Church was implying the *coextension* of the three concepts. Gödel, Robin Gandy (1919-1995), and Wilfried Sieg noticed that Church's statement was not sufficiently proved concerning recursive functions, whereas the other branch of the statement will become the *Church's Thesis*, namely that all effective procedures are λ-calculable, which is a statement for which Church never provided a proper demonstration[197].

Turing caused the two traditions on mechanical procedures to merge, since its idealized machine is nothing over and above an idealized human that does calculations step-by-step with a procedure that applies only a finite number of actions[198]: as I wanted to show, here we have both automated, mechanical calculation and ordered procedures at once. Turing explicitly describes his machine as such in both the introduction of his paper[199] and the passage in which he explains informally the proof of the extent of the computable numbers[200]. It is possible to see that his machine executes an *effective procedure in an informal sense, which his work allowed to be rigorously defined in terms of an idealized, theoretical mechanism,* described as follows[201], plus the five constraints that I introduced following Sieg in §1.1[202]:

---

[195] Soare 1996, pp.287-289.
[196] Church 1936, p.346. As it is noticed in Soare 1996, p.295, Church proposed this equivalence as a definition, not as a thesis.
[197] Soare 1996, pp.290-291.
[198] Soare 1996, pp.291-293.
[199] Turing 1936, p.231.
[200] Turing 1936, pp.249-252.
[201] See Minsky 1967, p.106 for the quotes in the list. All italics in the text.
[202] Cfr. also Soare 1996, pp.292-293.

1. an effective procedure is «a set of rules which tell us (my note: read instead "the machine"), from moment to moment, precisely how to behave»;
2. for any interpretation of the rules in question to be avoided, the following are demanded:
    a. «a *formal language* in which sets of behavioral rules are to be expressed»;
    b. «a *single* machine which can interpret statements in the language and thus carry out the steps of the specified process»;
3. The machine mentioned must have the features indicated in §1.1.

As Minsky notices, this definition is informal, yet it is rigorous[203], and in fact it was accepted by the highly esteemed mathematicians I mentioned, and it is still accepted nowadays[204], except a range of possible variations on those aspects that have a psychological background. Given the explanations provided above, *Turing's definition of being* computable (to be "computed by a Turing Machine") means: there is an effective procedure as defined at points 1 and 2 and performed by a machine which has the features recalled by point 3, which is "circle-free". So far, only the notion of "effective procedure" has been developed: a procedure "mechanical" in the sense of being unambiguously and completely specified by rules that define what to do for reaching the next result in the series of operation, i.e. an algorithm whose rule of operating is given, which we can call "γ" following Turing[205]. The concept "circle-free" still needs to be explained.

The notion of *being "circle-free" implies that the machine in question never stops*: it means that the machine prints infinite symbols of a first logical level[206]. Contemporary readers would expect the opposite: a machine able to fulfill its task is expected to halt, and it is exactly this halting that would prove the computability of a number. As McCormick notices, this is true only under the presupposition that the concepts "halting" and "being circle-free" are equivalent, but this is not the case: according to MacCormick, Turing did not define anywhere the concept of "halting", even if he discussed an equivalent problem[207].

Since Turing's definition need to be adapted to discussion of contemporary topics, I follow Minsky and integrate the halting condition, so that I adopt *Minsky's definition of being*

[203] Minsky 1967, p.105.
[204] See also Soare 1996, p.294.
[205] As in the passage quoted above from Turing 1936, p.239.
[206] Turing 1936, p.233:«If a computing machine never writes down more than a finite number of symbols of the first kind, it will be called *circular*. Otherwise it is said to be *circle-free*». Italics in the text.
[207] MacCormick 2018, p.322 and following.

computed by a Turing Machine: «*a function f(x) will be said to be Turing-computable if its values can be computed by some Turing machine T_f whose tape is initially blank except for some standard representation of the argument x. The value of f(x) is what remains on the tape when the machine stops*»[208]. This amounts to say that *every function whose values can all be enumerated through an effective procedure, executed by a Turing-Machine until its end, is "computable"*. In fact, "to enumerate" means that it is possible to list the elements of something, by saying that all those elements must be listed by a TM, one says that there is an effective procedure – implemented in the specific way of Turing Machines – whose complete execution provides a list of all the values of the function at issue. This will be our definition of being "Turing-computable" according to the previous reconstruction of Turing's proposal.

Since it lies at the basis of the claim that classical Computationalism is not committed to the idea that brain is organized as a Turing-Machine, it is important to notice that it is *partially* possible to abstract from Turing's *model* of computation, namely the model of the machinery that implements the effective procedure. The abstraction from the model allows to talk about "*being computable in general*", i.e. "being computable" as meaning "*being expressed as the result of enumerating by an effective procedure all the values of a function*". In fact, the notion of effective procedure and the alleged necessity of not relying on the interpretation of rules by stating them in a proper formal language – an aspect from which Turing abstracts, since he had not the problem of specifying the actual functioning of his machine – does not entail per se *what machinery* performs the effective procedure *and how*. Thus, it is possible to define further model of computation, since it depends on the existence of an effective procedure, i.e. according to the previous explanation, the possibility of enumerating all the values of a given function. Whether the machinery does it by an associative procedure or something else, whether it has a serial single control unit or parallel multiple control units, these are all aspects left undefined by the concept of "computing". Nevertheless, I said "partially possible" regarding this abstraction because "computing" has been defined as a "machine-like" operation in the sense specified by points 1 and 2, therefore both defining the suitable formal language and the mechanism that understands it and executes the relevant operations expressed through it are duties of a computational theory of mind.

Regarding the definition of "being computable", it could be asked what it is implied by Turing's demonstration that "all λ-computable sequences (functions or numbers) are

[208] Minsky 1967, p.135.

Turing-computable and that all Turing-computable sequences are λ-computable"[209]. Since the coextension of λ-calculus with general recursion had not been sufficiently proved by Church, since the hypothesis that λ-calculus covers all the effective procedures is just an inductive conjecture (Church's thesis), since the *Turing's thesis* says that «all the "computable" numbers include all numbers which would naturally be regarded as "computable"»[210] is a conjecture as well (even if it has been progressively confirmed), the following statements hold:

1. being "Turing-computable" means to be *effectively enumerable*, i.e. it is possible to list all the values of a function (say γ with Turing) for which it is possible to define an effective procedure – a procedure that can be carried on with mechanical means, understood as specified – that can be completed;
2. all that is Turing-computable is also λ-computable and the other way around;
3. Church stated that all effective procedures are λ-computable (Church's thesis), whereas Turing said that all procedures that are intuitively effective are Turing-computable (Turing's thesis). Given the previous point, *the Church-Turing thesis* declares that:

   3.1. λ-computation and Turing-computation have the same extension;

   3.2. all effective procedures are both λ-computable and Turing-computable;
4. *only according to Church the statement in the previous point would imply "all the effective procedures must be recursive"*, but given that Church's work regarding this passage was not accepted by the community of mathematicians, it follows that *it is only a conjecture following from Turing's thesis that all general recursive functions must be Turing-computable*, given that general recursive functions are effective procedures; nevertheless, "being computable" by itself does not entail "being recursive"[211];
5. notice that there is a constant reference (A) to the *mechanical* way in which the machine operates, and (B) to being written in some sort of formal language. These are all *constraints on being computable*. Ignoring them leads to serious misunderstanding, as well as ignoring the relations listed here. Some of these possible mistakes also affect philosophy of mind, and Copeland 2004 offers an interesting survey of some of them made by famous authors. For example, regarding the relation between "computing"

[209] Turing 1936, pp.263-265.
[210] Turing 1936, p.249.
[211] Soare 1996, p.312. Soare also points out that "being recursive" does not entail "being decidable", which means "there is the possibility of defining an effective procedure for showing whether or not $x \in M$ in some sense" (x being a property, a set or otherwise). For this definition see amongst other Berto 2008, p.30. Berto points out also that every enumerable set is also decidable (Berto 2008, p.32).

and the *model* of TM, as Fodor noticed, classical Computationalists are not committed to say that the brain has the functional architecture of a Turing Machine; nor more in general that every computing, concrete machine must be Turing-computable or *necessarily* no computing at all[212]. As in the case of Sloman's, Samuels' and Aizawa's remarks, caution should be adopted for not overestimating nor underestimating the importance of Turing's work and the extent of his contribution to the field.

1.3. The Theory of Universal Turing Machine

Minsky describes straightforwardly what a Universal Turing Machine is: :«[…] there exists a certain Turing machine that can imitate the behavior of any other Turing machine, given an adequate description of the structure of that other machine. The description has to be written down in the environment – tape – and need not be built into the works of the universal machine»[213]. Informally, the principle of a Universal Turing Machine (UTM) is quite simple: there is a TM that can compute any function "$\boldsymbol{M}(\alpha)$" that gives as outcome the series computed by the Turing Machine $\boldsymbol{M}$ whose description number is α. Minsky's description also provides *the two key features of both theoretical Turing Machines and concrete devices*:

1. the set of instructions, arguments (variables) and the alphabet in which those things are written must be contained within the tape (memory) of the TM[214], in such a form that machine can both access and "understand" them;
2. the behavior of a TM is determined by instructions and data, but the UTM simply computes its operation over the D.N. of a TM, in such a way that the two machines provide the same list of outputs.

There are two implications of point 2 that are worth to emphasize. First, it is important to remember the difference between the two definitions of function given above and fix a notion of *equivalence*. Since a function $f(x)$ is said to be Turing-computable if there is a TM which prints all its values, two TMs are *equivalent* if their function $F(x)$ *defined associatively* cannot be distinguished for all t when they are set in motion at the same moment on the same arguments[215]. In this case, the functions are also said to be equivalent.

[212] See Copeland 2004, p.106 and following.
[213] Minsky 1967, p.112.
[214] See Turing 1936, p.242: «the manner of operation of *M*' (my note: the UTM) could be made to depend on having the rules of operation (i.e. the S.D.) of *M* written somewhere within itself (i.e. within *M'*); each step could be carried out by referring to these rules».
[215] Cfr. Minsky 1967, p.134. See also Priestley 2011, p.90: «*U* is a Turing machine which can be supplied with the machine table for any other Turing machine *T*. It works through the steps that *T* would have performed, recording on its tape all the details of the configurations that *T* passes through in the course of a computation. Matters can be arranged so that *U* writes down exactly the output symbols that *T* would write

Second, the fact that UTM can compute the same series of TM by computing its D.N. shows that *instructions and data are treated in the same way*, or as Priestley puts it: «*there is no a priori difference between data and instructions; the distinction between the two is established only by their interpretation*»[216]. In fact, remember that which I said about the formal calculus in Frege: syntax is constitutively the only way for distinguishing symbols for variables from symbols for operations.

Less obvious for people who are not in the field is *how* to construct mathematically and concretely such a device. That which I am going to do in the rest of the chapter consists of relating the theory exposed so far and the functional and mathematical features of a Universal Turing Machine to concrete devices, so that a more comprehensive introspection in the classical computational theory of mind (CCTM) is provided: once it is understood precisely how a computational device works and what "to compute" means in this context, it will be demonstratively showed what "to computationally represent" means. As I already clarified through the survey of Sloman's, Samuels' and Aizawa's remarks, the reader must not be afraid that my conclusion is that a representation and/or the operations that implement it must be expressed in Turing-computable terms for CCTM to be consistent. "The plot is thicker".

There are different ways of constructing mathematically a Turing Machine, even if the principles are the same. Turing 1936 provided his own; Hermes 1965 and Minsky 1967 propose other versions of this mathematical construction, different from that introduced by Turing. Hermes' version is interesting, since it applies recursion to define the main operations of UTMs and applies recursive function to the D.N. of imitated TMs, but it goes excessively technical for our purposes. For a detailed account, I send to Priestley 2011, which also clarifies the original notation in Turing 1936. I provide only a general overview.

The first passage consists of writing the values of the function φ of the TM to be imitated (from now on "targeted TM") on the tape of the UTM. For the UTM to work, there are two conditions to be met:

- UTM must read all possible Q_{TM}, S_{TM} and D_{TM}, hence these sets must be *translated* into the corresponding sets of UTM, since it is not possible to posit that they are the same in every occasion, except D_{TM}, whose members are fixed by hypothesis for all TMs[217], moreover the operations of TM and UTM must have the same

down, even though the internal details of the computation carried out are different».

[216] Priestley 2011, p.119. Italics in the text. See also Eigenmann and Lilja 1999, p.7 for the same thing regarding the Von Neumann architecture.

[217] Cfr. Minsky 1967, p.139: «we have to use some representation like binary numbering to designate the

effects, even if they are not executed in the same order. This condition mirrors a basic principle of computers, that is *all programs must be translated into a code that has a standard form*, that is one compatible both with the operations and the syntactic rules of the machine that implements the operations at issue. This corresponds to the operation of *compiling* in concrete devices;

- a proper syntax must be provided. This mirrors the operation of *loading a program into a memory in a proper order*, which is one of the preliminary operations that a computer must execute before running a program[218].

The progress of any TM is defined by its instructions (the *m*-configurations and the actions and motions they prescribe), the symbols on the tape and the position of the scanner, since it determines the scanned symbol in the next operation. That which a UTM needs to individuate is the position of each of those variables on its tape[219], so that UTM can retrieve each of these pieces and act accordingly, that is it associates each configuration with each outcome and the next configuration. Thus, basically *UTM computes associatively the m-function* $\varphi(q_{ij}, s_{ij}, d_{ij}) = q_k, s_k$ on its tape by associating specific positions of its memory.

In fact, first, the current configuration is marked, and then an effective procedure is applied for finding the instruction coherent with the marked configuration, which once found is marked accordingly. A "check-point" is executed (which is another effective procedure) that confirms the correspondence between the two marked strings. This check is possible since both the instruction and the configuration must have the same beginning once they are written on the tape of UTM, where configurations (q_i, s_j) and instructions that complete the definition of transitions $(q_i, s_j, q_{ij}, s_{ij}, d_{ij})$ are listed separately. If they do not match, the UTM repeat again the search after the markers are erased, if they do, then the UTM goes on and individuate the next pair that form the configuration on its own tape. The process is repeated until the whole complete configuration of TM is marked, then another effective procedure prints all the outputs of the computation executed by TM, and all marks are erased on the tape of UTM. Thus, UTM has its own algorithm, but the same outcomes

states of the machine *T*. The reason is that the universal machine *U* which we are constructing is just one fixed machine, hence it is limited to some fixed set of tape symbols. On the contrary, the machine to be simulated, while each one of them is finite, may require arbitrarily large numbers of state, so *U* cannot afford a different symbol for each *T* state».

[218] Cfr. the operations of the program "*loader*" in Patterson and Hennessy 2013, p.129.

[219] Cfr. Priestley 2011, p.91: «the progress of *T* is defined by its table, by the symbols written on its tape, and by the position of the currently scanned square. In general, these last two factors will change at every step of the computation. *U* therefore keeps a record on its tape of the complete contents of the tape of *T* at each stage of the computation, along with the standard form of *T*'s table». By "table" the author means a list of the values of φ of the targeted TM written vertically with all the elements of the same set in the same column.

of TM[220]. This corresponds to *a general-purpose computer that executes a program written in a standard form*.

In the next paragraph, I illustrate the process of executing a program in concrete computers, so that the theory illustrated in these pages is related to concrete, classical computational devices. Once this has been done, I will discuss some consequences that are relevant for any CCTM and the notion of "representing" it wishes to apply, but that are not discussed, according to my experience of the literature.

§2. A Context for the Discussion of Von Neumann Architecture

First things first, allow me to provide a context for the description of hardware organization in contemporary computers that the reader is going to face. In fact, since ordinarily very few – maybe even no one – care about these details when discussing the CCTM, the reader could ask why I judge necessary to bother with such a detailed description for a proper philosophical understanding to be developed. My argument goes as follows.

The issue in the rest of the chapter consists of describing how concrete devices implement the ideas featured in Turing's concepts of "Turing-computable" and "Turing Machine", for the purpose of clarifying what a "computational representation" in the classical sense is. The reason is that by definition a "computational representation" must be implemented by a computation in some sense, and there are only *two* models of computing devices: serial processors[221] and computing connectionist architectures. Thus, following the line of reasoning developed by Ramsey, a "computational representation" *must be* something "computed" in a comprehensible sense by either one or the other of these devices, at least given the current progress in relevant sciences regarding "computation" and "computational devices".

At the moment, I confine the discussion to classical computation, and those devices compute *algorithms in the form of internally stored programs*. Thus, my interest lies in discussing how programs are executed *for a twofold purpose*: (A) showing how classical computers are organized further clarifies what "computing" means in concrete machineries; (B) that which they compute shows how they represent, since *if there is a representation, it must be one of the things computers compute/implement*. If this is the case, then the second enquiry allows also to understand properly what kind of reference a

[220] For this reconstruction, see Priestley 2011, pp.307-316.

[221] Serial architectures actually are not entirely serial: the processor executes one instruction for each cycle, scanned by an internal clock, this is the reason for which it is said to be "serial" and the speed of processors is measured in hertz, but actually many processes are run in parallel

computation can have. Thus, to sum up, *the first enquiry "A" concerns the problem of how a representation is implemented and why it is "computational", the second "B" concerns the problem of what a representation is and what kind of reference it can afford for itself insofar as it is defined as "computational"*.

There are of course some presuppositions in this argument. First, it seems that it must be clarified what a representation is before individuating the item that works as a representation in computing devices. I just stick to the literature, and maintain that "computational representations" can be only two kinds of items, corresponding to two families of computational architectures: physical signals that play the role of symbols by partaking in formal languages in serial computing devices, meaning that those signals stand in such a causal and operational relation that they mimic the operations of the machinery as they are expressed by programs; and physical patterns whose distribution mimics a certain understanding of external events/states of affairs through causal relations and properties of operations they perform[222]. I do not question if these are representations at the moment: this is the duty of *Chapter Fourth*.

The second presupposition is apparent: it could seem I think that if there is a semantic content, it is situated in programs: "that which is computed/implemented". This is not the case. I discuss the implementation of programs because they are the functional description of the activity of computing devices: they specify that which is computed and how at a functional level, namely they describe the flow of the alternation of physical states within physical structures in a form that makes more explicit that which they accomplish. Given what a "representation-vehicle" is considered to be, it seems that if there is a semantic, it lies in the fact that those physical states follow a transition that has the meaning (i.e. can be interpreted as it is) described by programs, and that such processes develop according to this meaning; therefore, I discuss the implementation of programs from the perspective of understanding *if these processes are just described as regular or if they are rules-following*. In both cases, my hypothesis is that semantic contents are ascribed to *processes*,

[222] See Egan 2019, p.247: «the representational vehicles in so-called 'classical' computational systems are *symbols*, physical structures characterized by a combinatorial syntax, over which computational processes are defined. Symbols are tailor-made for semantic interpretation, for 'hanging' contents on, so to speak. But not all computational systems are symbol-manipulating systems. For example, connectionist models explain cognitive phenomena as the propagation of activation among units in highly connected networks; dynamical models characterize cognitive processes by a set of differential equations describing the behavior of the system over time. The systems so described do not operate on symbols in any obvious sense. There is a good deal of controversy about whether these systems are genuinely representational. For the most part, the dispute concerns whether such systems have representational vehicles, that is, states or structures causally involved in cognitive processes that are plausibly construed as candidates for semantic interpretation». Italics in the text.

not to programs, that instead just clarify what this meaning is. Thus, when I say that "something that is computed/implemented" must be that which represents, I do not exclude the possibility that *the process itself is that which it is supposed to represent*; quite the contrary: I affirm it, since I think that there is no room for doing otherwise, because programs as the user experiences it are pure abstractions, they already suffer the difference between "being" and "being described as" that I stressed previously.

The view that semantic content is intrinsic to computational process qua computational is the core ground of the so-called "semantic conception of computation", a thesis according to which semantic content is intrinsic to computation, in the sense that it is not possible to define a computational process without referring to its interpretation and/or to explain its development unless one refers to the interpretation of the process[223]. At the end of the chapter, I aim to show that this view neglects a fundamental condition of "representing", namely both the item used to interpret and the interpreted representation-vehicle must be given to the owner of the representation together with their connection into a representation as explicit data, or there is no representation at all *for the owner of the representation* – which maybe cannot even said to be a representation any more.

From the standpoint of discussing the semantic view of computation as a crucial thesis for understanding import and legitimacy of classical computationalist theory of mind, the two analysis I proposed to carry on – what is computed and how it is computed within classical computing architectures – attempt to solve the problem of making clear what that which is computed is for the computer, and either the computer can access its state in the form of representations, or it does not, and if the latter is true, computational processes (the flow of state transitions) are not representations, since I will argue that there can be representation only if there can be a proper subject that relate specific items with specific contents and possesses the connection as an explicit datum, at least potentially. About this, the reader must not think that my thesis is that computer would require phenomenal consciousness as such in order to represent, my point is that they cannot satisfy the *formal structure of a representation*, which I will discuss in *Chapter Fourth*.

Over than the criticisms in *Chapter Fourth*, here in §§6-7 I start to prove the claim on the missing feature of representation in computing devices by showing that the content to be used for interpreting the symbols computationally manipulated, so that they are symbols

[223] Crane 1990; Piccinini 2006 proposes a variant that attaches semantics to mechanical implementation by arguing that the functional role – which is semantic – is individuated by the role of the underlying mechanism; Shagrir 2018. Jerry Fodor was defending this kind of view too.

according to the concept of "symbol"[224], is beyond the reach of the processes themselves, hence there are no symbols at all in computing devices, there are *signals* that external subject with symbolic competence can *describe as* symbols. From the standpoint of computing devices, there are *informative signals*, a concept I will clarify in §§6-7 of this chapter. I show that the flaw at stake is already implied by the concept of a formal language. If I am right, there are not even the fundamental parts of a representation in a computational process per se, in that which *it is*; therefore, a representational view of computation is doomed to be an extrinsic view on computation, which cannot ascribe semantic content *to computational process*, it can be *only one way of understanding and/or describing them*.

§3. The Von Neumann Architecture, Part 1: Organization and Components

It is acknowledged that the beginning of modern computers is due to the idea of "stored-program computers"[225], that is the idea that both instructions and data are registered in the memory of the computer[226], in the same form, so that there is no *a priori* difference between the two, as Priestley emphasized concerning Turing Machines. In modern computers, the "stored-program" concept is realized by representing both operations and operands by numbers through binary code[227]. Behind this "representing" there is the following situation: programs are stored in memory cells that are flip-flop switches with two values, namely on and off, which correspond to the two signs of a binary code and actually are different electric signals.

As I showed in the previous paragraph, the idea of a stored program, is embodied in the concept of a Universal Turing Machine, and other historians have also defended this thesis[228]. Nevertheless, it is with EDVAC that this concept received an explicit definition and a concrete implementation. There is a debate about who had this idea within the team that was projecting EDVAC, and maybe the name "Von Neumann Architecture" for its functional organization excessively emphasizes the contribution of John Von Neumann (1903-1957) to the project, since he was mainly concerned with the logical aspects of automated computation, but the big issues of implementation were faced by the directors of the project, Mauchly and Eckert[229].

[224] The reason for which researchers should care that symbols applied in computational processes are symbols is the same for which they should care that "computational representations" are actually representations according to Ramsey 2006.
[225] Priestley 2011, p.126, Patterson and Hennessy 2013, p.49.
[226] Patterson and Hennessy 2013, p.49.
[227] See Patterson and Hennessy 2013, p.66.
[228] Priestley 2011, p.129.
[229] Priestley 2011, pp.126-130.

Notice also that the *First Draft* discusses the idea of a stored-program, general-purpose computer by an extensive use of metaphor regarding computers as brain, and it seems that Turing played some role in framing the debate in such terms[230], even if he was not much involved within the development of the EDVAC project, since he was already working for the British government[231]. Other sources for the idea of relating brains and machines are McCulloch and Pitts' work on logical neurons[232] and Bigelow and Wiener's work on feedback, who also stresses the similarities between mechanical behavior and organic behavior[233].

In order to discuss the main topic (execution of stored programs) in proper terms, it is necessary to give some definitions and an overview of the organization of the active part of the computer, the processor. All that the machine can execute is the *machine language*, that is the binary code, whose alphabet consists of two kinds of electric signals[234], that *are represented* by the famous 0 and 1. The two forms of signals are defined respectively "false" or "deasserted" signal and "true" or "asserted" signal[235]. Instructions in the machine language constitutes strings of the *machine code*. Above the machine language there is a first class of symbolic languages, that is *assembly languages*, a very simple code that can be translated into binary code. The *program* that translates assembly languages into machine language is called "*assembler*" [236], whereas the *program* that translates high-level languages (e.g. C++, Java, Python, LISP and others) into assembly language is called "*compiler*" [237]. By "high-level languages" it is meant some language that can be compiled in a comprehensible form for a computer, which is written in some natural language or mathematical code such that it mostly abstracts from the concrete operations of the machine[238].

Except for the machine language, so far only software has been discussed. The hardware that implements the software performs four basic *functions*: inputting data, outputting data, processing data and storing data[239]. My concern is mainly with the last two functions. Processing data is performed by the "Central Processing Unit", commonly called

[230] Priestley 2011, pp.130-131.
[231] Priestley 2011, p.133.
[232] Priestley 2011, pp.132.
[233] Priestley 2011, pp.131-132.
[234] Patterson and Hennessy 2013, p.14.
[235] Patterson and Hennessy 2013, p.250: «we will use the word **asserted** to indicate a signal that is logically high and *assert* to specify that a signal should be driven logically high, and *deassert* or **deasserted** to represent logically low». Bold and italics in the text.
[236] Patterson and Hennessy 2013, p.14.
[237] Patterson and Hennessy 2013, p.15.
[238] Cfr. Patterson and Hennessy 2013, pp.14-15.
[239] Patterson and Hennessy 2013, p.16.

"processor" or CPU. The CPU is divided as follows[240]:

1. *datapath*[241] or "*datapath elements*"[242]: it is the set of components that performs arithmetic and logical operations. There can be multiple datapath, each composed as follows[243]:
    a. "Program Counter", or "PC": it is «a register that holds the address of the current instruction»[244]. It means that there is a memory built (commonly) directly into the processor which holds the code that corresponds to a certain location in the memory of the computer outside the CPU that holds the instruction to be executed[245];
    b. the block of memory internal to the CPU that holds instructions and data to be executed;
    c. "Arithmetic and Logic Unit", or ALU: it is the net of "logic gates" that implements arithmetical and logical operations[246]. Logic gates are embodied operators of Boolean logics and arithmetic, they are structures hardwired in such a way that the signals they release as outputs in consequence of certain inputs mirror the truth tables of the mentioned operators[247]. They are described logically according to the convention previously defined;
2. *control unit*: it is «the component of the processor that commands the datapath, memory, and I/O devices according to the instructions of the program»[248]. Otherwise stated, the control unit is the component that collects input signals that check the state of other components, and provides output signals that set the state of other components in order to perform properly the instructions contained in the memory;
3. *memory registers*: these are locations of memory inside the CPU, they are devoted to common operations and to function as working memory for the operations the CPU is called to execute. Registers are flip-flop switches hardwired in such a way to retain their state, independently of what signals have sent them, unless the *writing line* enables the process. A writing line is a line arranged in such a way that the configuration of the

[240] Patterson and Hennessy 2013, p.19.
[241] Patterson and Hennessy 2013, p.19.
[242] Patterson and Hennessy 2013, p.251.
[243] See Patterson and Hennessy 2013, p.251 and following.
[244] Patterson and Hennessy 2013, p.251.
[245] Patterson and Hennessy 2013, p.16.
[246] Patterson and Hennessy 2013, p.B26.
[247] Cfr. Patterson and Hennessy 2013, pp.B4-B8. By "truth table" it is meant a schema that shows how the truth or falsity of a proposition varies with the variation of truth and falsity of its variable terms.
[248] Patterson and Hennessy 2013, p.19.

switch can be altered when it is asserted. At a level of abstraction above the hardware, a memory register can be described as a string of propositional calculus that is always true or false once a certain value is fixed, unless the variable corresponding to the writing line is asserted. *This is what a memory is into a computer in general*. The length in bit of information that a register can contain is called its width[249];

4. *clocks*: operations in the computing machine must be synchronized, especially reading and writing operations, which are involved in defining the series of inputs and outputs of the operations. For this reason, computers are provided with a clock, a special chip where a vibrating material is stocked, and its vibrations and their interaction with signals scan the time of the operations[250].

As a last preparatory note, keep in mind that «the organization of the processor, including the major functional units, their interconnection, and control»[251] is called *microarchitecture*. The rest of the definitions and details concerning organization of components are provided step by step when their knowledge is required for understanding the fetching-decoding-executing cycle.

In the next paragraph, we will see that the challenge of defining programs is the challenge of conceiving complex operations in terms of operations and means that the microarchitecture can receive after a proper translation[252], and in such a form that the machine can execute it. This implies that computing machines have within their own "intellectual reach" – that which is actually given to them per se, and they actually handle – only the flow of signals into the microarchitecture: *neither representations, nor procedures as such, not even symbols*. As in the case of Turing Machines, all that is available to the machine is: its state, the inputs, an associated behavior that leads to an

[249] Cfr. Patterson and Hennessy 2013, pp.B50-B57. See also on memory in general Patterson and Hennessy 2013, p.248: «other elements in the design are not combinational (my note: such that they provide a constant output for constant inputs), but instead contain state. An element contains state if it has some internal storage. We call these elements state elements because, if we pulled the power plug on the computer, we could restart it accurately by loading the state elements with the values they contained before we pulled the plug. Furthermore, if we saved and restored the state elements, it would be as if the computer had never lost power. Thus, these state elements completely characterize the computer. […] the instruction and data memories, as well as the registers, are all examples of state elements»

[250] See Patterson and Hennessy 2013, pp.B48-B49 for a detailed account of the topic.

[251] Patterson and Hennessy 2013, p.347.

[252] This does not mean that the challenge of programming consists of conceiving in terms of microarchitecture the purpose one wants to implement. It means instead that ultimately all the complex tasks that computer can do reduce to the implementation of a set of basic operations in a proper order, and that this relentless mechanical form is all that computers "understand" – and this is not a representation, unless one wants seriously loosen such a concept at the point it does not make any sense applying it, as Ramsey suggests. Unlike Ramsey, it is my claim that classical model of computation is not representational too. I know this may sound very radical, but I will do my best to clarify the point and present (I hope) convincing arguments for this.

output – and they are available as *implicit conditions (states), not as representations*, declarative or whatever else. In fact, the development of the procedure in each moment is defined by state-transition formulas as (indefinitely more complex, but the same in principle) those I explained in §1 of this chapter. Thus, programs/procedures are not considered as independent objects, i.e. they are not understood as procedures independently from their application. Even when computing machines retrieve the whole program, this is done just because a specific syntax has been provided, and proper markers and constraints have been fulfilled in the construction of a program that corresponds to physical and hardwired features: there is no such a thing as the consideration of the procedure in the form of explicit datum.

§4. The Von Neumann Architecture, Part 2: Executing Programs

As I said, computing machines can handle only sequences of instructions in physical binary code, and there is a standard cycle of operations that takes place independently of the program being executed[253], namely the "fetching-decoding-executing" cycle. This cycle is preceded by the procedure of *compiling*: the procedure that transforms high-level languages manipulated by users into the physical correlate of the machine code. *Four programs realize this process*; they are: compiler, assembler, linker and loader. They correspond to as many steps of the compiling process, and they operate in the order in which they are listed here. My analysis will not go in many technical details: the point is understanding that there is no such a thing as the program we experience as users of computers[254], and clarifying what a program is for the machine, in order to clarify in turn what may be a reasonable *referent* of "computational representation".

Of course, my assumption for the conclusion I am going to defend is that there is no such a thing as a representation as an absolute object, namely I reject that there can be representations independently from the relation of the representation-vehicle with a suitable subject, hence if there must be a "computational representation" qua representation, necessarily it must consists of something that is given to the "computational subject" *exactly in the form that it is given to him/her/it*. A complete account of this view is provided in *Chapter Fourth*, but in the meanwhile, I just postulate this point for the sake of exposition. Provisionally, I can argue through a famous example. Hilary Putnam in

[253] Patterson and Hennessy 2013, p.251.

[254] Patterson and Hennessy 2013 emphasize in many places (for example pp14-16) that high-level programming languages are means to enhance the capacity of programmers and users of conceiving instructions (program engineering) and even hardware (see for example the pages in Appendix B dedicated to logic gates). Thus, programming languages *are implemented descriptions in terms of formal languages* of underlying processes.

Reason, Truth and History makes the case of an ant that involuntarily portraits Churchill on the sand. Is this a representation? My point is: it is a *potential* sign, but it is not *in actuality*; for it to be an actual representation, it needs a relation with a proper subject. The point is that the sign *does not refer per se to its referent*. If so, the sensation of the object should provide also the interpretation, but this is not the case: in fact, if one allows that the sign is an actual representation, this means that the sign per se has reference, but in this case, since there is no subject who interacts with the sign, reference must depend on the existence of the sign as a thing; in turn, this implies that given the sensible existence of the sign – the only objective existence that is appropriate to call for here – interpretation should be provided too, but this is opposite to experience, hence representation is the result of an activity of the subject on the thing.

Let us return to the main exposition. Another thing that the previous discussions have showed is that the difference between operations and operands is purely syntactical. Thus, the indication of these information must be specified by instructions. In fact, each instruction of machine code is divided in *fields*[255], that specify the locations of operands and of the portion of memory where the result of the operation must be stored[256], together with what the operation to be performed is and the length of the shift in the allocated (read as: "previously managed distribution of") memory. The fields are not constant: they may differ according to different kind of instructions[257], but of course fields that are implied by the operation to be performed must always be present. For example, there must be an *opcode*, that is the field that specify the operation to be performed for given arguments. Opcodes are not limited to logical or arithmetical operations, also basic operations as "load" or "write" into memory addresses can be triggered with proper opcodes.

The preparatory operations to the fetching-decoding-executing cycle (FDE cycle from now on) are the following:

- a *compiler* translates the *source language*, namely the high-level language in which a program is written[258], into the assembly language. This is that which compilers do in general: for each program, a compiled version of the code is provided, or even a compiler designed precisely for the given language;
- an *assembler* turns the compiled code into an *object file*, that is a binary code that

[255] Patterson and Hennessy 2013, p.61.
[256] Notice the correspondence with a UTM: it executes any procedure by locating the operations, operands and the succession of states within its own memory.
[257] Patterson and Hennessy 2013, pp.82-83.
[258] Patterson and Hennessy 2013, p.A6.

reports instructions and data together with the description of how properly collocate them in memory, in such a way that variables and operations have consistent references[259];

- both compiling and assembling are run in parallel: several processes run at the same time, so that each procedure that forms a program is compiled and assembled independently. This allows to manage corrections and repetitions into the program with less computational resources. A *linker* connects all these parts into a single file; moreover, its part called *link editor* checks the correspondence of the references within the portions of the assembled program and resolve undefined variables. Its duty is also to specificy the actual memory addresses that will host the parts of the programs, as indicated during the assembling stage. The linker produces *an executable file*, that is a file that is ready to be handled by the control unit[260];
- assembling provides an assessment of memory size and position of all the parts of the object file (object file header)[261]. The loader reads this portion of the object file and generates an address space suitable to contain it, then it starts to copy it according to the instructions elaborated by the linker and set memory locations for starting the program[262].

I begin with the assumption that the program has already been placed into the memory in its compiled form, and that the program counter has already been set to the memory address of the first instruction. This allows me to initiate immediately the discussion of the FDE cycle. The presupposition does not hide any relevant detail, since the compiling process is a program that is executed as any other through the FDE cycle I am going to describe. According to the explanations provided so far, the fact that the program is loaded in the memory means that a series of memory blocks has been set in such a way that the discharging of their signals in the proper sequence *can be described as* a certain series of instructions and related arguments expressed into the binary code.

The figure 1 is the functional schema of a CPU, including registers, Control Unit, ALU and the related hardwiring. The clock is not indicated, since for the present purpose is sufficient to know its function of synchronizing the flow of signals. The trapezoidal figures like the ones marked with “add” (it means “adder”) and “mux” (it stands for “multiplexor”,

[259] Patterson and Hennessy 2013, pp.107-108.
[260] Patterson and Hennessy 2013, pp.108-109.
[261] Patterson and Hennessy 2013, p.108.
[262] Patterson and Hennessy 2013, p.112

see below) indicate nets of logic gates. Thus, those elements concretely are chips wired in a certain order so that their outputs mirror logical or operational properties.

Amongst the components, it is specifically important to have a proper understanding of the complementary concepts of "multiplexor" and "decoder", hence I provide a detailed explanation below. The importance of these two classes of components and the related concepts lies in this: *they form the core of the "understanding" and the "capacity of comparing and distinguishing" of the computer*. Thus, they play a central role in selecting which "intelligent" operation to do and in operating "choices". According to my understanding of the topic, multiplexors and decoders are the hardware base that is supposed to explain the rationality of operating in computers, since they are responsible for elaborating instructions and trigger the proper execution. The reason for this claim will be clarified and confirmed with the exposition itself.

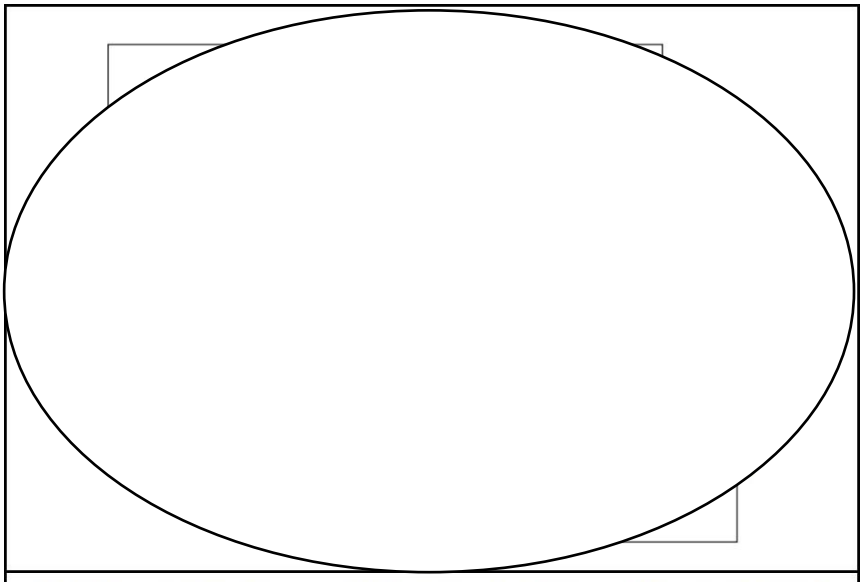

Fig.1. Schema of a CPU, with control and ALU pipelining from Patterson and Hennessy 2013, p.265. I don't possess any right on this image. All the rights belong to their owner in the original text.

Multiplexors and decoders have both the function of selecting data, and *functionally speaking* it could be said that they *identify* certain inputs and produce suitable outputs for triggering consistent responses or procedures. Nevertheless, here we start to appreciate the depth of the difference between "being describable as" and "being" something, which I mentioned above. For there is no such a thing as an "identification" in the proper sense, namely, to recognize something *as something else* – in fact, here this would be the sense to be applied, since the function of these components consists of individuating some occurrence as a case of something else, which is a form of "identification"[263]. In fact, I am going to show that multiplexors and decoders exert the function of selecting data from

[263] When a human identifies something, *a difference is produced*. When for example it is said "*x* is a portrait", when I recognize that the object in front of me is a portrait, the *content* of my cognition – whatever it may be its form, namely discursive, neural or what one prefers – is nevertheless as follows: the sensible item, which is *x*, is a case of "portrait", whose expanded form is that it is an artifact with a certain function and so on. This content implies *a difference between two items to which a different logical role is assigned*: namely substrate and predicate.

several alternatives, so that different functions directed to the same component can be coordinated[264]. Notice that the word "selecting" must not induce the image of just one line sending a signal: *all lines are either asserted or deasserted*. The point thus is asserting the correct one.

As it is showed in fig.2, the hardwiring of the MUX is as such that only if the signal line is in a certain state, a correspondent input is transmitted. Otherwise stated, the logical design of a multiplexor (which is illustrated in the truth-tables of the circuit), is such that only if a certain signal line is active or not one of the alternatives is selected. This has two implications: first, logically speaking, multiplexors correspond to the Boolean operator "material implication"[265] (but the reader can see they *are* not, this is a circuit, not an operator, nowhere the computer can access this operation as individuated in such terms); second, for *n* inputs, a multiplexor must have $\log_2 n$ selecting lines, since for each couple of inputs there must be a single selecting line, as it is showed in the figure. If the number is odd, it is sufficient to add one line for selecting the option.

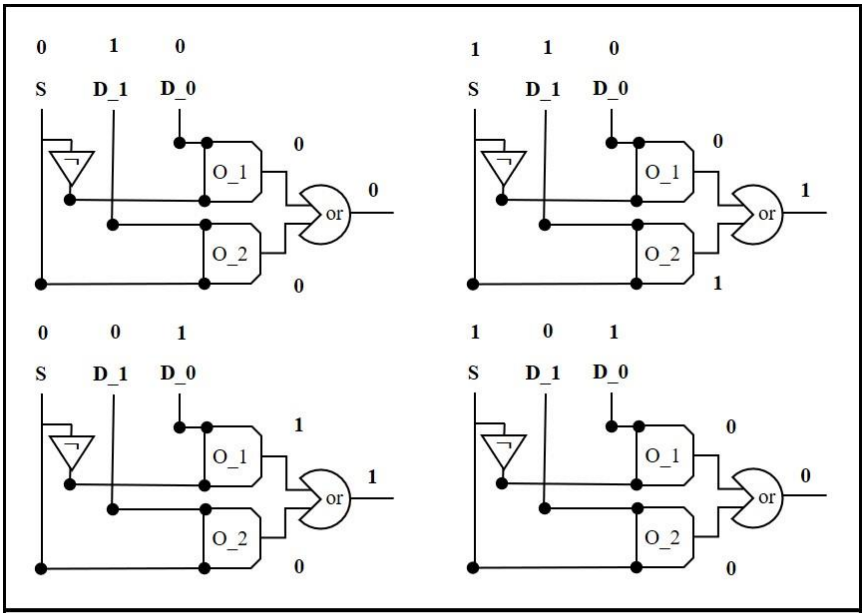

Fig.2. Schema of a multiplexor. Dots represent intersections. The figures on the right represent 'and' gates, whereas the triangles on the top stands for lines that transmit the opposite value of the main lines. The circular figures are 'or' gates. If the reader compiles the truth-table, he/she can see that the multiplexor always transmits the value of the line corresponding to the value of the selection line S.

Now I am going to discuss structure and functioning of a decoder. The reader can consider fig.3 for a visual support. The duty of a decoder is also to select one amongst many alternatives, but conceptually there is a difference: whereas the MUX is logically corresponding to "$s_x \rightarrow d_x$", the decoder must be represented as "$A_x \wedge B_y \wedge C_z \rightarrow O_k$". Thus,

[264] Patterson and Hennessy 2013, p.246: «the value written into the PC can come from one of two adders, the data written into the register file can come from either the ALU or the data memory, and the second input to the ALU can come from a register or the immediate field of the instruction. In practice, these data lines cannot simply be wired together; we must add a logic element that chooses from among the multiple sources and steers one of those sources to its destination. This selection is commonly done with a device called a *multiplexor*, although this device might better be called a *data selector*». Italics in the text.
[265] Patterson and Hennessy 2013, p.B10.

the MUX transmits a certain choice by transmitting the value corresponding to a certain line, whereas the decoder transmits a certain set of choices by asserting a certain line. As the reader can appreciate in fig.3, the decoder is as such that there is a set of possibilities for a certain number of input, each of them representing one of possible configurations, for example one of the opcodes from a finite set of possible codes. If a signal standing for that operation is transmitted, then only the channel corresponding to that signal is asserted. As it is showed in the figure, for each *n*-bits width of the input, 2^n outputs are provided, since as many the alternatives are, and as I said, for each line one is asserted and the others deasserted. In such a way, a decoder can signal what input has been provided by asserting the corresponding line and deasserting the others.

Notice that, while I was giving the explanations of MUX and decoders, I used terms from the semantic family of "representation", and I spoke of "standing for" and similar. A supporter of the semantic view – for example, one who uses arguments similar to Shagrir 2018 – may indicate this as an example of necessity of a representational characterization of computational devices and conclude that this indicates a contradiction in my thesis. For the sake of completeness, allow me to reject the objection from the very beginning: if one shares my premises on the impossibility of acknowledging representation as an absolute object, the cases now examined must be conceptualized as correspondence (in this case caused by intentional design, but even if one calls for another perspective this makes no difference), not as "representation", since this is how humans describe these items, but the computer only does that which its internal mechanism is designed to do. Sure, this must not be understood as meaning that a computer is like a light switch, nor the lesson to be drawn is that all of this is smooth and obvious – all the opposite – it is just that this is a non-semantical process, in all its steps, no matter how mandatory a logical description is *for humans* for this process to be properly understood *by humans*, since they are not the subject needed for the computer to be a subject able to represent in a proper sense.

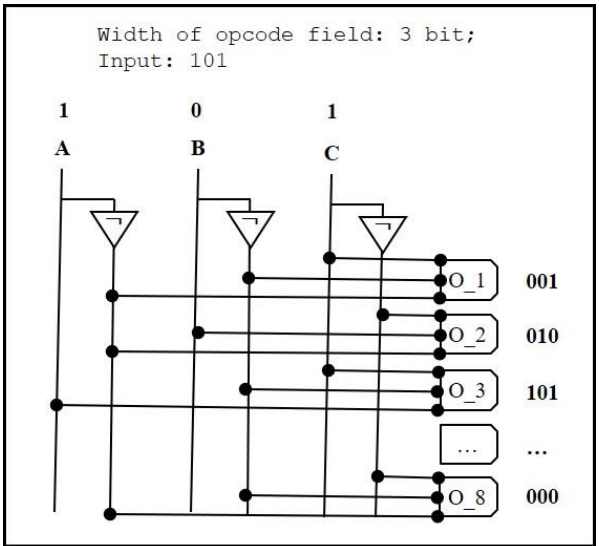

Fig.3. Schema of a decoder. Dots represent intersections. The figures on the right represent 'and' gates, whereas the triangles on the top stands for lines that assert the opposite value of the main lines. If the reader compiles the truth-table, he/she can see that only Output3 (O_3) matches the input.

Now that these steps have been clarified, it is possible to begin the exposition of the FDE

cycle. The cycle at a sufficient level of abstraction is relatively simple[266]. First, as I explained before, the program counter has the address of the first instruction loaded within. This means that the block of memories is set in such a way that their ordered transmitting communicates a signal that can be described into binary notation as the number corresponding to a memory location. Memory in fact is arranged in blocks as American cities: each street runs parallel to the others, so that they form a grid, and when one has to give indications, he/she indicates the crossroad[267]. The corresponding field of instruction acts in the same way. Notice that «for every instruction, the first two steps are identical: (1) send the program counter (PC) to the memory that contains the code and fetch the instruction from that memory[268]; (2) read one or two registers, using fields of the instruction (my note: i.e. the fields containing the opcode) to select the registers to read. For the load word instruction, we need to read only one register, but most other instructions require reading two registers»[269].

Notice also the PC is part of a feedforward, cyclic structure, as it is visible in figure 1: when it transmits the number of the address, the same value is transmitted to an adder, which increments the register to the value of the next instruction. In this way, the next instruction can be fetched once that the process has been developed, since the loader has already set the memory values in such a way that this constant operation correctly fetches the next instruction.

Once the instruction has been fetched – which means that the CU has received signals from the PC that allowed a reading at a given address through some operation in the ALU, which is thus already been appropriately set by the CU for this purpose[270] – the instruction fetched is written into the instruction memory register, which is an internal working memory. At the same time, the field of the opcode in the instruction fetched is sent to the CU, which checks what kind of instruction is and what states of the other components it requires for it to be executed[271]. This is possible because of two main factors:

---

[266] As it is explained in Silc, Robič and Ungerer 1999, there are many different ways of organizing the steps of the process, and others have been invented after the year of publication of this book. For example, Patterson and Hennessy build the theory on the MIPS architecture, which is used in processors like Intel-i7 and other widespread models. My interest is in a functional description, that accounts for the steps of the FDE cycle. How they are implemented in each model makes no difference for the argument I am developing.
[267] Patterson and Hennessy 2013, pp.B58-B66.
[268] Read as follows: "use the program counter (PC) to address the memory location that contains the code and fetch the next instruction from it".
[269] Patterson and Hennessy 2013, p.245.
[270] Reading and loading instructions use the ALU as well as computing instructions. Cfr. Patterson and Hennessy 2013, p.245.
[271] Patterson and Hennessy 2013, p.265.

- *microarchitecture compatibility*: the purpose of compiling, as I said, it is to translate the high-level language into machine code, that is series of signals that can be described through the binary code. This operation is needed because instructions must become "comprehensible" to the machine, but what does it mean this in practice? How machines relate to their instructions? *Behind the word "understand" there are engineering factors*. The first is the compatibility of the microarchitecture with the instructions provided. When we want to refer to the microarchitecture insofar as certain resources are needed to implement a given set of machine instructions, we call it "instruction set architecture" (ISA)[272]. Different microarchitectures can implement the same set of high-level instructions, if they are compatible with the instruction set[273]. Notice that in the theoretical approach of TMs the problem is ignored (legitimately, since Turing's purpose was to set mathematical concepts and procedures): the fact that "L" is associated with "move one square to left" is left unspecified in the paper, TMs just act according to the shape of symbols" in the same way as a *human* that has to calculate some proposition through formal notation does;
- *the role of decoders*: Control Unit is the actual responsible for the interpretation of the instruction. One important step in interpreting an instruction is understanding *what kind of opcode* the instruction brings, so that it can be determined what the system must do with the data supplied: writing, reading, executing arithmetical operations and so on. The previous constraint of being translatable into a finite instruction set of high-level instructions warrant that there is a limited number of values that the CU can receive as opcodes. Without this constraint, there would be an unlimited number of options, which would cause the necessity of defining an unlimited number of states, so that the state-transition function φ could not be defined, both for its extent, which would cause a proliferation of mnemonical needs and hardwiring, and for the reasons explained in §1.2. Since instead there is the limitation indicated, a suitable *decoder* can be designed. The intervention of the *control decoder* – the decoder dedicated to decoding the instructions from the instruction register – marks the beginning of the *decoding phase* of the FDE cycle.

[272] Eigenmann and Lilja 1999, p.390: «the collection of instructions is called the instruction set, and, together with the resources needed for their execution, the instruction set architecture (ISA)».
[273] Patterson and Hennessy 2013, p.86: «the commercial implication is that computers can inherit ready-made software provided they are compatible with an existing instruction set. Such "binary compatibility" often leads industry to align around a small number of instruction set architectures».

Similar operations are performed for every field of each instruction. Once the instruction has been decoded, the CU sends corresponding *control signals*[274] to other components, as it is showed in figure 1. The CU can generate almost all the control signals from the opcode[275], and it generates «[…] a write signal for each state element, the selector control for each multiplexor, and the ALU control»[276], as it is shown in fig.1. The presence of an ALU control is part of a strategy of multi-layered decoding: «this style of using multiple levels of decoding – that is, the main control unit generates the ALUOp bits, which then are used as input to the ALU control that generates the actual signals to control the ALU unit – is a common implementation technique. Using multiple levels of control can reduce the size of the main control unit. Using several smaller control units may also potentially increase the speed of the control unit. Such optimizations are important, since the speed of the control unit is often critical to clock cycle time»[277]. From the explanation of how decoders and multiplexors work, the reader can appreciate that setting the control and then operating according to instruction is a purely mechanical matter, one that is heavily relying on structural features of design and physical interactions.

Once that instructions are decoded and the components are set into a proper disposition, the *phase of execution* begins. The CU sends signals that activate the registers, from which signals are orderly sent towards the components of the ALU, which in turn are set in such a way that, again, because of their material arrangement manipulate the signals so that their operations can be described as the proposition of the original high-level formal language in which programmers and users operate. The coordination of discharging and setting is grounded on the clocking system.

Once that the instruction is performed, the result is written into memory, and another cycle can begin. Of course, I described the model on a linear way for the sake of exposition, but

[274] Cfr. Patterson and Hennessy 2013, p.250: «a signal used for multiplexor selection or for directing the operation of a functional unit; contrasts with a *data signal*, which contains information that is operated on by a functional unit». Italics in the text.

[275] Patterson and Hennessy 2013, p.263: «the control unit can set all but one of the control signals based solely on the opcode field of the instruction. The PCSrc control line is the exception. That control line should be asserted if the instruction is branch on equal (a decision that the control unit can make) *and* the Zero output of the ALU, which is used for equality comparison, is asserted. To generate the PCSrc signal, we will need to AND together a signal from the control unit, which we call *Branch*, with the Zero signal out of the ALU». Otherwise stated, it means that every signal is generated by the CU except the one that enables writing in PC when there is an instruction that send to a memory location which is not consecutive to the one written in the PC. The "Zero signal" is a control signal that the ALU applies for checking that the value in the PC is identical with the value calculated by the ALU for the next instruction. The MUX in the up-right side of fig.1 selects if the value of the line from the PC or the one calculated by the ALU will be transmitted, according to the mechanism I explained before. For more details on this step of the control, see Patterson and Hennessy 2013, p.247.

[276] Patterson and Hennessy 2013, p.257.

[277] Patterson and Hennessy 2013, p.260.

many steps can be run in parallel, for example multiple decoders are allowed, and of course many processors. Moreover, one has not to be afraid that this way of operating requires thousands of instructions to perform even the simplest operations: a processor with a speed of 1GHz is able to perform 10^9 cycles *over each second*.

§5. The Conclusions Following from the Previous Paragraphs

The purpose of the previous exposition consists of showing that:

1. As TMs, contemporary computers are behavioral computing machines, namely they implement the execution of functions through implementing ordered procedures, that *are described* as series of formal propositions manipulated according to some propositional calculus within a formal language, but this is not that which they are for the machine. I have shown that every execution presupposes compiling, that is a work of transcription of the series of inputs in the high-level language form, and according to the high-level language rules, into a simpler code. This is possible because their compatibility is already coded and presupposed – compiling itself is a procedure that is already expressed into the machine code and automatically set. I have also shown that once this operation has been completed, then there is a mechanism that in a purely mechanical way sets other components to perform in an equally mechanical way *information processing* that corresponds to the high-level instructions provided at the beginning. Thus, it is possible to see that information processing is purely behavioral, since it consists of ordered series of causes-effects that are entirely classifiable as an action-response course of events, in which, as in the theory of TMs, the next state and action is entirely determined by the previous state of affairs and the incoming inputs;
2. there is no such a thing as interpretation and representation for computers and their operation. The superimposition of content, that is the distinctive feature of any interpretation (a premise I provisionally defended with the argument in §4 of this chapter) is absent, and all the references to the distinctive features of languages – syntax, semantics, grammar and so on – reduce ultimately to non-semantic and even non-linguistic items: namely, structural organization and control hardwiring. By consequence, as I argued, unless one is willing to consider the property of having reference relatively to representation-vehicle as an absolute property and therefore representations as absolute objects, it is illegitimate to posit representations in both states and operations of classical computers. Computers and computational devices are very peculiar automated artifacts, they cannot be equated to linear mechanisms without neglecting the fundamental features of control and self-control, together with the

consequences of feedback causation (memory in circuits for example heavily relies on this) and the fundamental role of logical and mathematical properties. Despite of this, neither they can be considered devices that handle representations in some meaningful sense: they implement processes that can be formally (mathematically or linguistically) described, whose description in turn provides a *sense* that *refers* to real or simulated objective processes and states of affairs. But from the standpoint of computational devices, all that is given is a state transition, and without the connection of signs with the content that has the function of interpreting, explicitly posited as such and explicitly given to a subject, it is hard to say that these "computational subjects" possess a representation or manipulate some, and that the set of signals or the succession of signals is a representation;

3. the semantic view of computation has some good reasons on its side, but on the light of that which classical computational devices are, that which "computing" and "computation" mean properly speaking, I think these reasons are applied to the wrong conclusion. As it results from my exposition too, it is impossible to claim that these devices can be properly understood by describing underlying mechanisms, for example as in the case of thermostats, motor engines, or similar classical mechanisms. It is instead necessary to refer to their function: if one checks classes (many courses are available online, both by professionals and amateurs) or the main handbooks for computer design, it results that the logical conception of components is unavoidable for realizing them, for making them do that which they must for making a working engine. Thus, the conclusion can be legitimately drawn that those devices compute: they actually do that, since stating otherwise make them unaccountable, and this is counterfactual, an objection that is heavier than usual since these are artificial products. The question that remains open therefore is: does computing implies representing? "Computing" means "to calculate the values of a logico-mathematical expression through an effective procedure, namely a procedure expressed within a formal calculus that follows step-by-step rules that leave no rooms to arbitrary interpretation and completely specify every passage". It is an action that is defined over calculus, propositional or not; there is room for interpretation regarding *how* the effective procedure is performed, since this is left unspecified by the concept "effective procedure" per se. Thus, claiming that something computes and that something represents are different claims, unless one is willing to state that calculating implies representing. Let us discuss the point. Of course, computation implies manipulation of symbols (indeed, in a particular sense, but this is something I will discuss only in the

next paragraph), and it may seem that this is enough for ascribing representations, but it is important to consider the explanation in §1. Turing showed the minimum approach to computation: the purely behavioral and associative approach by finite means, which is not only possible, but even highly rewarding, and this is a matter of fact that computers exemplify clearly. A finitist approach like Turing-computation is thus always possible, and it is the minimum and therefore *sufficient* constraint for computing. If I am right in considering Turing's definition of computation the minimum common denominator of models of computation, "computation" in general implies *symbolic discrimination*, not *symbolic competence*. By the former I mean the capacity of distinguishing symbols, by the latter I mean the capacity of *constructing and/or interpreting symbols*, that is another thing: the second implies the first, but the opposite is not true. In fact, TMs just sense signs, and distinguish them for operating accordingly, since the "computer" (the one which computes) must associate different actions to different symbol in a constant way, but this does not require necessarily symbolic competence, as the model of TMs shows, as computational devices themselves concretely show, and as it follows with sufficient evidence by definition and history of the concept of "computing". Thus, "being able to compute" and "possessing representation and/or operating with them" are *two different issues*. At the very best, one could claim that *some* models of computation imply representation, but neither Turing's, nor concrete classical computing devices, nor connectionist ones (as I am going to show) provide sufficient ground for this attribution, hence it is unclear which case should allow such connection between the two concepts of "computation" and "representation".

4. there could be a possible objection: is it not sufficient that there must be symbols for computing in order to claim that there must be some kind of symbolic competence, at least for manipulating symbols qua symbols? Sure, "for… *qua symbols*", i.e. only *if symbols are manipulated as symbols*, but they can be manipulated as material items exploited as *stimuli* for selecting a proper *action*. In fact, classical computers do not fit the condition of manipulating symbols qua symbols, as I showed. For something to be a symbol, at the very least it must be something that stands for something else, but again, unless the objector is willing to allow "representing" to be an absolute property and "representation" an absolute item that coincides with the sign-vehicle, then this property must become actual when the item is received by a proper subject, but in this case the proper subject possesses only a mechanical architecture that grants a

differential response[278] that *corresponds to* symbolic discrimination, but *there is no room* for ascribing a symbolic competence.

These four points form the premises of further conclusions on the issues I raised in §3 and on the main topic of this work. I state them in the following pages, then I move to another problem regarding the possibility for computational devices of being representational, which is self-contained, hence I can discuss it separately. The conclusions that follow from the exposition so far are:

A. "*to compute*" means to find an effective procedure in order to enumerate all the values of a function, namely a procedure that is rule-governed in such a way that each step is exhaustively defined, and no arbitrary interpretation is possible. It implies:

   I. *only a set of operations can be said "computable"*: according to the Turing-thesis, all the functions and numbers that are naturally regarded as computable are computable by an effective procedure, and so far all effective procedures have been proved to be Turing-computable, hence all *and only* the Turing-computable functions and numbers are computable, namely those whose values can be defined by the finite means of effective procedures understood as Turing-computation. If one wants to be more general and not get committed to the hypothesis, "computable" in general must be said to be all *and only* those functions and numbers whose values can be enumerated through an effective procedure in general defined over a formal calculus. A computational theory of mind in general – independently from how defines the implementation of computation – must thus be able to show that *every* mental operation follows an effective procedure at some level of organization *defined over a formal calculus*, just because it claims that *x* is "computational". A "computational representation" must be a representation that is achieved through the computation of the values over a logico-mathematical structure, operation that is developed according to an effective procedure;

   II. thus, computation implies that *effective procedures must be defined*, but *at some level of organization in the execution*. As I showed, "computation" is defined over Turing's formalism, even if its concept is looser than its theoretical and concrete implementation. *This does not imply that an effective procedure must be explicitly*

[278] Ramsey 2006 acknowledges to classical devices that they represent, since they manipulate symbolic items in a clear sense, but complains that the differential responses of connectionist architectures are not genuine computations. Piccinini 2008, pp.315-316 points out that often opponents of connectionist computation underrate the fact that classical computers are ultimately nets of logical gate, which compute the relevant logic operators over strings of digits (signals), therefore either at least some connectionist architecture performs computation, or nothing does. My view is closer to Piccinini's in this case.

possessed by the subject that computes: it is required that it acts accordingly, and that its operations have as arguments logico-mathematical structures and rules. Adding further constraints would be illegitimate. Regarding the possible objection that my description would like to ascribe an effective procedure without acknowledging any explicit possession of rules and arguments expressed in an actual formal language, see the pages on information, especially §§6-7. Information in Shannon's sense allows to fulfill the requirement of computing as executing an effective procedure over a formal language, but without ascribing a counterfactual linguistic competence in a representational sense;

III. computation presupposes *a logico-mathematical structure of that which the computation is operated over*. Since it is a mathematical operation, a suitable argument must be provided.

It does not imply:

I. that *only Turing-computable arguments*[279] *are computable*. Even under the widening assumption that all the functions and numbers that are naturally regarded as computable are computable by an effective procedure, and so far all effective procedure have been proved to be Turing-computable, all that the concept of "computing" implies is that an effective procedure must be specified *in general* for something to be said "computable". The concept of "computing" per se leaves undefined how the effective procedure is found, performed or implemented. The point is that the effective procedure is *defined*, and that its results and arguments are mathematical. About this, it is important that executing a program and computing are not equated, not because all programs are not effective procedures, but because the latter (executing an effective procedure) per se does not imply the former (executing a program in serial architectures)[280]. Executing an effective procedure can, more in general, be equated to executing an *algorithm*, but this holds because it is ultimately tautological, thus consider this as a remark about use of words;

II. about the previous point, notice also that *nothing prevents an effective procedure from applying probabilistic considerations*: the procedure must be effective, namely it must specify the steps of the process and reach a result – if any – with finite means in a finite amount of time, but there is nothing against the hypothesis

[279] Read as: numbers and functions that are Turing-computable.
[280] Cfr for example Piccinini 2008, p.315.

that the state-transition is non-deterministic, even if this is not a classical Turing-computation. In fact, Leeuw, Moore, Shannon and Shapiro in their *Computability by Probabilistic Machines* (1956) quoted Turing, Kleene, Church, Post and Davis for the concept of "effective procedure", they claim that their results concern recursion theory, which we have seen with Soare to be a case of effective procedure, and they show that there is an effective procedure to compute computable numbers with probabilistic means;

III. that *computation implies semantics*. I regard this as not true, for at least two reasons. The first concerns the implementation of computation in computers and, as I am going to show, in connectionist computing devices as well. Computers are information-processing devices: all they handle is a flow of binary code whose strings stand in a meaningful relation between each other, but this "being meaningful" depends on mechanical properties, as I have been showing, hence there is no way of ascribing semantics, since the linear succession of actions and chains of stimuli-responses cannot be equated to the act of superimposing items over other items, in which I claim interpretation consists – a thesis that I will fully defend in *Chapter Fourth*. The second reason is directly related to the theory of computation. Turing's definition, as described in the first paragraph of the chapter, is the standard definition of the concept, and it prescribes just the possibility of defining an effective procedure, whose concept does not imply any reference to "representation", since a behavioristic approach that deals with functions in a pure associative manner is always possible, because TMs are exactly this, as I showed in §1 of this chapter. Thus, claiming that "computation implies representation" amounts to say that defining or maybe implementing an effective procedure would imply to recur to some representation, but given the possibility of a behavioral and associative approach to performing computations, and the lack of evidence in concrete cases that a different approach is implemented, it follows that the claim "computation implies representation" is conceptually original over than counterfactual. Fodor embraced this thesis, and he credited Turing with ignoring the problem of symbol and content in computation and with including them *as semantic items*[281]. This is not Turing's point: Turing takes for granted the ability of *symbolic discrimination*, but nowhere he mentions a *symbolic competence of the machine*. I already showed that: when Turing speaks informally of "state of mind",

[281] Fodor 1998, pp.11-12.

he equates the reference of this expression to instructions that "tell what to do". Since TMs operate by reacting to symbols they scan in a pre-ordered way, they associate actions to symbols, not meanings, and this is not an act of representing. The problem is that *there is not another definition in computer science over than Turing's* (as I showed in the discussion of Soare 1996)[282], and since "computing" is a technical notion, that makes sense in its own context – moreover, Ramsey's epistemological remarks still hold – researchers should stick to Turing's definition, which is not Fodor's. Claiming that computation is an operation that respects semantical relations, as Fodor does, is misleading. This is not nominally false: as I said, it is impossible to account for computing devices without referring to logical and syntactical properties at least, and it is legitimate to state that logical items as Boolean operators mean something, therefore have semantic value and therefore, from the standpoint of the operator, components as logical gates represent (*stand for*, but do they also refer – send – to them?) logical operators/operations. Nevertheless, this does not imply that *there is* semantic value in a computation per se, nor this view is supported by concrete examples of computing devices, unless it is stated that semantic value for items is an absolute property: from the standpoint of computing subjects, these operations and components *stand for* logical operators/operations, but they *are* just the flow of information and of internal states, they are *courses of actions*, not symbolic productions;

B. *clarification on the definition of "computation"*/"*computational*". Thus, on which basis is something defined "computation" or "computational", according to the exposition so far? As Piccinini 2008 remarks, ascribing computation on the ground of executing programs is not a viable option, since this would contradict the matter of fact that both some artificial neural nets (for example recurrent neural nets[283]) and some classical

[282] For the sake of accuracy, according to Rojas 1996, pp.5-9, the mathematical definition of computation as "recursion" and Turing's computation as effective procedure (practical model of computation), Von Neumann's model of computation (computer model of computation), the cellular automata model (computation without separation of memory and processor, as in ENIAC) and the biological model of computation (that of artificial neural network) should be considered *five different models* of computation, whereas I argued that the first is a definition of a set of effective procedures, but not all of them; the second and the third definition of computation are in principle one and the same, and they differ only in terms of theory-practice; the fourth has been dismissed and the fifth is the alternative to the second-third definition. In rejecting this view, I just followed the scientific literature on the topic, and the historical pattern that Soare reconstructs. In fact, I have reported that Gödel acknowledged Turing's definition as the correct definition of effective procedure and computing, and I have showed in the previous paragraphs that Von Neumann architecture is the practical realization of a TM under relevant respects. Moreover, no one is even mentioning anymore the fourth model of computation: it disappeared from scientific handbooks, and Rojas himself just mentions it. For all these reasons, I confirm the statement I am commenting.

[283] Cfr. Graupe 2007, p.233. "Recurrent neural nets" are nets such that the output of each layer is an input for each of previous layer, and the process of feedback produces the weighting of connections within discrete

computers can be described by the same family of formal languages[284]. As I was saying in the point (a) of the list about what is not implied by the concept of "computation", there is not just one way in which an effective procedure can be executed, and it is important that the effective procedure can be defined over than executed. Since at least some artificial neural networks can be described with the same formal tools applied to describe classical computers, there is room to state that they can at least be described as computing devices. In fact, this was ultimately one of the premises for ascribing computation to classical mechanisms, together with the necessity of referring to logical descriptions and concept in order to account for such mechanisms. Artificial neural networks commit to the same necessity: how they work is unaccountable without understanding that they perform iteratively a certain set of functions[285]. This necessity provides a ground for ascribing the act of computing to connectionist devices, independently from the relation to a subject. Thus, *my point is that "computation" and "computational" are definitions regarding an activity, and if this activity provides a sufficient ground for the necessity of being conceptualized as computation and this ground concerns absolute features of the subject who performs the operation* – in this case, structural features of implementation for classical computers, functional-mathematical features of the operation for artificial neural nets – *then an operation is objectively a computation*. Otherwise stated, my conclusion is that computation consists of executing an effective procedure over elements of a logico-mathematical language, hence "computing" and "being computational" can be ascribed for every subject or operation that necessarily must be described in this way on the ground of absolute features of the performer or the process, where "absolute" means "not dependent on the perspective or the descriptive framework adopted". As I said where I was discussing the point of a semantic view of computation, this implies that such artifacts or subjects are performing a computation in proper sense and that this is an objective statement, but this does not imply that those subjects and items *represent* something, nor this implies that they possess a language and operate over that language in an ordinary sense, as I am going to show in §§6-7. The difference between the facts that *for us* computational artifacts such as computer performs computations and that

cycles. The fact that such networks can be described as Turing-complete – namely they can compute all functions that a TM can compute – since they are equivalent to finite-state machines (TMs belong to this set) has been demonstrated in Rojas 1996. Turing-completeness of McCulloch and Pitts neural nets, which Rojas uses as basis for defining recurrent neural nets, is demonstrated also in Siegel and Sontag 1991 and 1995.

[284] Piccinini 2008, p.312.

[285] This is showed in the relevant scientific handbooks, as Rojas 1996 or Haykin 2009: the properties of artificial neural networks depend on the mathematical features of the functions they implement.

the same computational artifacts have no clue of this, must be accommodate as follows: computational artifacts can be described as sign-vehicle of computation from our standpoint *when a relation of representing between us and such items has to be affirmed*[286], but computing systems are computational by themselves since the accomplishment of a computation and the property of these devices – regarding both their structure and their operations – do not depends on our account of those items, nor on our perspective over them.

As a closing remark of the paragraph, notice that the points $A_{II\text{-}III}$ in the list has been regarded by themselves as sources of troubles for Computationalism: Dreyfus' criticisms for example are centered on the impossibility of even defining a complete series of rules for generating every human behavior[287]. Since this objection is old and it seems to require further clarification and data for it to be properly debated, I just point out those issues, but I do not assume them as objections in my work.

In the rest of the chapter I discuss a further problem. It is commonly acknowledged that a representation is transmitted through a sign, that is some material trace that sends somehow to something different to itself. According to that which has been explained, the sign that form computational representations are the inputs and outputs, and I showed that they correspond to lines that are asserted or deasserted, namely to series of signals that stand in a meaningful relation between each other. The signals then should be the physical vehicles of a formal calculus, and the fact that computational devices act accordingly to rules described into a formal language should depend on the fact that those signals are occurrences of a formal calculus: their being meaningful consists of this.

Even if such a thing as a signal were a sign, representation and even just the formal language qua language (like these signs I typewrite here) would stand intrinsically beyond the reach of any formal system, so that they become impossible to be specified, i.e. *explicitly declared* in some form. This is of philosophical interest, since the problem is the same of finitist mathematics, which aimed to solve mathematical problems without any appeal to conceptual understanding, through a purely behavioral attitude towards formal schemas. Computers have the same approach towards computation, a Wittgensteinian one, in which behavior (allowed uses) decides the meaning, and the former ultimately reduces to the latter. Frege was the first to question Hilbert position on this problem. The review of

[286] "Representing" is a perspective on a certain object, again because otherwise we are compelled to consider "representing" as an absolute property of thing as things.
[287] See the case in Dreyfus 1968.

the debate between Frege and Hilbert by Cassirer can shed some light on the problem of intrinsic interpretation of a system that handle only information as signals.

§6. The Problem of Sign in Relation to Information Theory: Historical Roots

6.1. Defining the Problem of Sign for Computational Devices

Claude Shannon (1916-2001) published *A Mathematical Theory of Communication* in July 1948[288]. In this paper, the topic of transmitting signals between telecommunications devices is translated into a mathematical form that has applications beyond the original field, an extension that the author himself foresaw. Shannon's main topic was thus communication engineering, and only in a successive moment his theory became integrated in the field of computer science, mainly through Wiener's *Cybernetics, Or Control and Communication in the Animal and the Machine* (1931, with further clarification in the second edition in 1961 and in a paper of 1948) and Von Neumann's idea of automaton (1948, but after Wiener, and of course Shannon)[289].

The strength of Shannon's theory lies in its versatility: as Aspray puts it, «the importance of this characterization (my note: Shannon's characterization of a process of communication) is its applicability to a wide variety of communication problems, provided that the five components[290] are appropriately interpreted. For example, it applies equally well to conversations between humans, interactions between machines, and even to communication between parts of an organism. Properly interpreted, the communication both between the stomach and the brain and between the target and the guided missile could be seen as examples of a communication system. Using Shannon's theory, previously unrecognized connections between the biological and the physical worlds could be unmasked»[291]. The reader can now guess what is the point in applying communication theory to computer science given this hint and the survey on the structure of computers in the previous paragraphs: it is a theory for making rigorous the coordination between the components of the computing devices, and for defining their exchange of signals in terms of exchange of content[292]. Wiener discussed information relatively to communication between the biological components involved in processes and the importance of feedback exchanging of unambiguous state messages for a proper control of biological functions to

[288] See the heading in the version of Shannon's paper I quote in the references, as Shannon 1948/1993.
[289] In Von Neumann 1951.
[290] The author refers to information source, transmitter, noise of the channel of transmission, receiver and destination.
[291] Aspray 1965, p.123.
[292] Cfr. Von Neumann 1951, pp.6-7;

be achieved[293]. Von Neumann instead discussed information in terms of how undesired fluctuation in the transmission of signals that regulates the internal behavior of the computing machines can be limited, so that they can be made more reliable and less prone to malfunctions[294].

Thus, according to Wiener as well as to Von Neumann, the point in applying information theory to biological and computational systems consists of defining how the different components of those systems can inform the others of their state and/or of providing consistent inputs that allow a coordination in order to perform complex operations and functions. Therefore, *signals are the means of this communication, functionally speaking, they have the same function of linguistic signs* (in the semiotic sense[295]). If this is the case, for the same reasons I asked through Ramsey's reflection whether there is any genuine representation in computational processes, it is possible to ask if there are genuine signs within computing machines, given that signals as information-vehicles work as signs in computing devices. This is of interest not only by itself, but also because the outcome of the enquiry concerns both classical and connectionist computational devices, and knowing if there are the basic constituents of representations may shed light on whether there are representations (supposed that all representations presuppose some sign, which I argue for in *Chapter Fourth*).

6.2. The Contribution of Nyquist and Hartley to Shannon's Information Theory

Before Shannon's proposal, other models were formulated for making mathematical the problem of transmitting semantic messages, but their application was limited to telecommunication, and there were flaws within this application too. I briefly revise those models before discussing the meaning of Shannon's definition of information, so that it will be clearer how Shannon's concept has to be interpreted.

---

[293] It is significant the application of the problem of control to psychopathology in Wiener 1948 and in Wiener 1961/1985.

[294] Von Neumann 1951, p.6: «the guiding principle without which it is impossible to reach an understanding of the situation is the classical one of all "communication theory" – "the signal to noise ratio". That is, the critical question is this: how large are the uncontrollable fluctuations of the mechanism that constitute the "noise", compared to the significant "signals" that express the numbers on which the machine operates?».

[295] Barthes 1964/1986, p.34 explicitly mentions that a signal for cybernetics is a sign in semiotic sense. I have no space available for a complete account of the relation between the Saussurean notion of sign and Barthes' definition of "semiological sign". I can only say that "sign" in the semiotic sense designates a trace in general – it can be an item, a practice, a sign in the common sense of the word, as for example a letter – which receives by its usage a significate. Sign is thus a twofold entity: "sign" designates both the material sign-vehicle and the signified content. The signified content and the relation of signification does not presuppose a discursive, linguistic nature: it consists of implications, relational attitudes, more or less explicit, as for example the possession of a certain car may indicate bravery. As Barthes puts it, signification is a *practice*. Cfr. Barthes 1964/1986, pp.34-51. The fact that signals are signs in the semiotic sense is that which I was alluding when I wrote that signals are signs "indeed in a particular sense". I am going to fully demonstrate this thesis.

The engineer Harry Nyquist published *Certain Factors Affecting Telegraph Speed* in 1924. Over than treating the argument announced in the heading, the author proposed a criterion for evaluating the adequacy of the structure of codes (languages) to engineering factors affecting the speed of transmission. The author was the first who thought of applying algebra for representing the material signals in the machine, but he articulated his algebraic conception on the basis of the structure of languages: the speed of transmission is measured according to the grouping of the elements, namely signals into characters, characters into words, words into propositions, propositions into messages. Nyquist can be credited for understanding that only the first level of organization matters for the engineering problem of communication, but he could not translate into algebraic relations the bounds of language at the above syntactic levels[296], so that the only factors considered were variety and length of characters, which grows exponentially with the length of communication. The measure is frustrating for computational purposes, since it would demand exponentially growing resources to be handled[297]. Moreover, the assumption of natural language as a model prevented applications beyond the original context.

These aspects can be traced in Nyquist's paper. The author calls his measure a quantification of «speed of transmission of intelligence» (*W*), defined as «the number of characters, representing different letters, figures, etc. which can be transmitted in given length of time assuming that a circuit transmits a given number of elements per unit time»[298]. The demonstration of the measure goes as follows:

- characters are supposed to have the same length[299], hence each character corresponds to a unit of time. Given that each signal has a constant length too, the total number of possible characters is m^n, where "*m*" designates the number of possible signals, and "*n*" designates the number of signals required for composing a character[300];
- since the system is in every moment identical with itself, m^n is constant (*K*) given

[296] For example, Nyquist's measure does not express facts as it is more common that a consonant follows to another consonant if the first is in the middle of a word, whereas is uncommon if the first is at the beginning of a word.
[297] It is not relevant to provide many details about this, just consider the following example. All serial computational machineries can handle only a finite number of digits. The common calculator is a perfect example: if a number is multiplied in such a way that the number of digits grows beyond the number of digits available on the display, an error message appears; this is a way of that machine of showing that this operation cannot be performed, since the exceeding digits cannot be considered. In the case of communication, a law as Nyquist's would cause very fast to incur in this problem. For the question in the example in technical detail, see Minsky 1967, pp.26-27.
[298] Nyquist 1924, p.333.
[299] Nyquist 1924, p.342.
[300] Nyquist 1924, p.343.

m and *n*;

- the number of possible characters increases fast, therefore the author expresses this number through a scale, that is $n \log m$. This is legitimate because both m^n and $n \log m$ are monotonic functions (they both only increase or decrease), therefore there is a bijective relation that allows to map one function onto another;
- let *s* be the number of signals that can be transmitted during a certain unit of time: this is the speed of the channel of transmission. Under the previous assumptions, the speed of transmission *W* necessarily is directly proportional to the width of the channel *s*, and inversely proportional to the length of a character, namely *n*. Since characters have the same length, they can be posited as a unit of time, hence speed of transmission can be defined independently from time. Thus, we have: $W = \frac{s}{n}$;
- in this form, there is no reference to the variety of signals and the rule of composition of characters (which are instead in the logarithmic form). Thus, we do the following substitutions:
  - $n \log m = K$ (it is constant for the reason specified above);
  - $n = \frac{K}{\log m}$;
  - $W = \frac{s}{\frac{K}{\log m}}$;
  - $W = \frac{s \log m}{K}$;
  - $W = \frac{s}{K} \log m$;
  - $W = k \log m$. Both *s* and *K* are constants, so they can be replaced by "*k*".

The meaning of Nyquist's law is that the speed of transmission is a function of the inverse relation between the width of the channel and the number of characters (which is *k*) and the variety of signals *m* in the language applied, which is called the "capacity of expression" of the language. Now that the equation of *W* has been explained, the remarks at the beginning of the paragraph can be justified. The relation *W* is still the relation between the width of the channel *s* and the length of the characters *n*, which is expressed as the modulation (in the mathematical sense) of the scaled variety of signals ($\log m$) by the constant relation between those two factors *s* and *n*. In this way, there is no information concerning the composition of the character, since *m* designates all the possible signals, hence *W* does not take into account the structural features of the language applied; nor *W* is informative regarding how communication develops, since the relation between *s* and *n*

is conceived as if the communication exploits all the possible resources of the channel for the whole length of the process. Nyquist's law is nevertheless useful because it shows a relevant consequence for computational purposes: Nyquist's law shows that there is little advantage in increasing the variety of signals used. In fact, the logarithmic function increases fast up to a threshold value, then it increases slower and slower even if constantly[301].

Nyquist's work is retrieved and deepened by another telecommunications engineer, Ralph Vinton Lyon Hartley (1888-1970), who published *Transmission of Information* in 1928. Hartley pushed Nyquist's work in the direction of a purely non-semantical consideration of the problem of communication, and this allowed him to formulate a different articulation, closer to engineering factors relevant for the problem of communication.

He divided messages into «selections»: the idea behind is that communication depends on the choices of some operator (of the device used for communicating) for each alternative item that composes the message, since each part transmitted excludes the transmission of alternative items[302]. Moreover, Hartley proposed to consider the problem of communication *from the perspective of devices*: that which matters in this case is to reconstruct the result of successive selections without ambiguity, or at least with the fewest possible ambiguity. Thus, the common mastery of a code and physical interferences become relevant problem. The former is regarded as potentially psychological, a dimension which Hartley wanted to exclude, hence it is presupposed that «[…] in estimating the capacity of the physical system to transmit information we should ignore the question of interpretation, make each selection perfectly arbitrary, and base our result on the possibility of the receiver's distinguishing the result of selecting any symbol from that of selecting any other»[303]. "Information" it is not a word that Hartley defines, but he says that it becomes more and more precise with the progression of selections, and he makes the example of "apples are red", which comments by saying that the further *information* of "the size of the apple" must be provided with further selections[304]. Thus, the reference of information is still the informal idea of "that which is communicated"; under this respect,

[301] Nyquist 1924, p.333: «it should also be noted that whereas there is considerable advantage in a moderate increase in the number of current values, there is little advantage in going to a large number».
[302] Hartley 1928, p.536: «for example, in the sentence "apples are red" the first word eliminates other kinds of fruits and all other objects in general. The second directs attention to some property of apples, and the third eliminates other possible colors. It does not, however, eliminate possibilities regarding the size of apples and this further information must be conveyed by subsequent selections». The selections necessary to compose characters are called "primary selections".
[303] Hartley 1928, p.538.
[304] Hartley 1928, p.536.

it is still equivalent to Nyquist's concept of "intelligence". Nevertheless, the way Hartley frames the problem of information is similar to Nyquist's in the shape, but the underlying interpretation changes, and this causes some advantages.

First, as I said, Hartley's measure of information quantifies the possibility of reconstructing the message by the receiver, which is a different problem from Nyquist's. Second, unlike Nyquist, Hartley considered that not all the non-redundant capacity of expression available in the language is at issue in the communication process, thus some changes to the model were required. Hartley says that «inasmuch as the precision of the information depends upon what other symbols sequences might have been chosen it would seem reasonable to hope to find in the number of these sequences the desired quantitative measure of information»[305].

The measure of the successive selection is constructed as follows. Each selection is composed of *n* choices and, supposed that each choice is arbitrary, the number of options available for each selection is s^n, where "*s*" designates the number of first-order signs of a language[306]. Thus, s^n is the number of possible strings of symbols given *n* choices. Given the compositional character of each language, from the set of first-order symbols S_1 is possible to compose a secondary symbol $x \in S_2$ each n_1 selections, and the process can be repeated at will. Since each selection is composed of independent, compatible events, it follows that $s_3 = (s_1^{n_1})^{n_2}$, hence in general, given "*i*" as the maximum order of symbols, it results that all the subsequent orders can be expressed as a power with base s_1 as $s_i = s_1^{n_1 \times \ldots \times n_{i-1}}$. The meaning of this is that Nyquist was wrong in assuming the composition of language as a model for constructing his mathematical definition[307], both because it is indifferent what the level of composition is, and because this measure grows exponentially in a discrete way, according to the values of set $\mathbb{N}$, so that the following criticisms hold[308]:

- it is implausible that the information *in Hartley's sense*, namely the being informative of the characters transmitted, grows exponentially at the proceeding of communication in such a way. Concrete communications have a course similar to the logarithmic function, in which an initial, fast growing tends to a threshold value and then become slower and slower informative, even if there can be peaks, for example the answers to yes-and-no questions to particularly significant;

[305] Hartley 1928, p.536.
[306] Hartley 1928, p.538.
[307] Hartley 1928, p.538-539.
[308] Hartley 1928, p.539.

- moreover, until the system is working properly, there is not an exponential growing of resources: there can be peaks, but for most of the time the demand of resources is constant.

Hartley draws the conclusion that «in order then for a measure of information to be of practical engineering value it should be of such a nature that the information is proportional to the number of selections. The number of possible sequences is therefore not suitable for use directly as a measure of information»[309]. Nevertheless, Hartley's idea consists of reforming Nyquist's law by abstracting from the capacity of the channel in order to consider just informativeness as a measure of the process of selecting. In fact, Hartley's law is: $H = \log_x s^n$ [310], where "*x*" designates an arbitrary base for the logarithm[311], "*s*" is the number of first-order sign available and "*n*" is the number of choices. Thus, let say for example that we are using the binary code and that the characters demand 5 selections to be composed, so that 32 characters are available as in the English alphabet, and set the base of the logarithm to 2, so that we measure information in bit, as Shannon will do. In this case, we have $H = \log_2 2^5 = 5$, namely each character is informative as 5, whereas because of our convention each selection is informative as 1, since it has 2 values and rules out 1 value. Notice the following:

- Nyquist's law expressed the speed of transmission as a function of the capacity of the transmission channel and the number of possible signs; here the value of being informative is expressed as a function of the number of possible choices for each first-order sign. In general, the basic unit of the first is the capacity of expression of the language, whereas the basic unit of the second is the variety of options for each choice that composes first-order signs;
- Hartley kept the logarithmic form of the relation, but he completely changed the underlying model: Nyquist's law measure an engineering feature, namely the width of the channel; Hartley's law instead measures an abstract feature, which was considered exclusively semantic, namely the being informative of that which is materially transmitted. Hartley's law is something in the middle between a physical and a conceptual measure;
- As Hartley remarks, «what we have done is to take as our practical measure of

[309] Hartley 1928, p.539.
[310] Hartley 1928, p.540.
[311] Hartley 1928, p.540: «the numerical value of the information will depend upon the system of logarithms used».

information the logarithm of the number of possible symbol sequences. […] If we put *n* equal to unity, we see that the information associated with a single selection is the logarithm of the number of symbols available [...]»[312]. In terms of our previous example, we have that $H = \log_2 2^1 = 1$. This shows that the measure of information consists in the logarithmic scaling of the number of possible choices. In this way, Hartley can keep the good part of Nyquist's work – i.e. the intuition of scaling the measure so that it does not grows exponentially – and give it a proper reason: the process of communicating becomes progressively less uncertain, i.e. less and less informative, since precision is the correlate of reduction of potential alternatives. At the same time, he improved Nyquist's work, as Aspray notices: «Hartley had arrived at many of the most important ideas of the mathematical theory of communication: the difference between information and meaning, information as a physical quantity, the logarithmic rule for transmission of information, and the concept of noise as an impediment in the transmission of information»[313].

The improvements indicated in the previous point will be the starting assumptions of Shannon's definition of information. There were still at least two missing points in Hartley's measure of information: first, even if *H* is defined by a material parameter (the number of possible selections), so that the questions concerning the mental side could be safely ignored, still the law cannot define the relation between the selections; second, it is still unclear how the measure *H* should make measurable also the possibility for the receiver of reconstructing the messages. Shannon's work will provide satisfying answers to these aspects too.

6.3. Shannon's Concept of Information

Shannon retrieves Hartley's approach to the problem of communication, namely he also considers the capacity of reconstructing a message at a given point as the right problem to be faced by a theory of communication[314]. Shannon also credited Hartley for individuating the logarithmic form of the law, even if any monotonic function would be suitable, and he praises the choice for three reason[315]:

- some important parameters vary in a comfortable way if the logarithmic form is applied. As in our previous example, where the base of the logarithm was 2, if one

[312] Hartley 1928, p.540-541.
[313] Aspray 1985, p.122.
[314] Shannon 1949/1960, p.31.
[315] Shannon 1949/1960, p.32.

wants to measure the expansion of information to a system by adding one flip-flop switch, then it just adds 1;

- the logarithmic form is consistent with intuitive assumptions;
- necessary mathematical operations become easier.

As in Hartley's paper too, the background assumption is that the actual message is selected from a finite set of possible messages, hence there is a finite set of choices[316]. Moreover, the choice of the logarithmic base corresponds to the choice of a unity for measuring, and Shannon introduces the famous *bit*, that stands for "binary digit"[317], namely each selection of a digit (in the mathematical sense) between two alternatives. We talk of digit especially for classical computational systems, whereas we talk in general of processing information for neural nets, since only the former have *discrete choices*, whereas the latter implement continuous states, even if these last too can have discrete states at a certain level of abstraction or under certain conditions, for example in the case of recurrent neural networks. This is possible since *information is defined for both discrete, continuous and mixed systems of communication*. By "system of communication" Shannon means the situation in figure 4, where the parts are defined as follows[318]: the *information source* is the maker of the message, the sender of information, and the message can be made of signals as well as one or more functions; *transmitter* is an encoder, namely some agent that makes the message suitable to be transmitted through the channel; the *channel* is the mean through which the message is transmitted; the *receiver* is a decoder, it performs the inverse operation of the transmitter; the *destination* is the subject for whom the message is intended. A *discrete system of communication* is one in which the message is transmitted by a sequence of discrete symbols; a *continuous system* is one in which the flow of signals is treated as a continuous function and a *mixed system* is one that includes both.

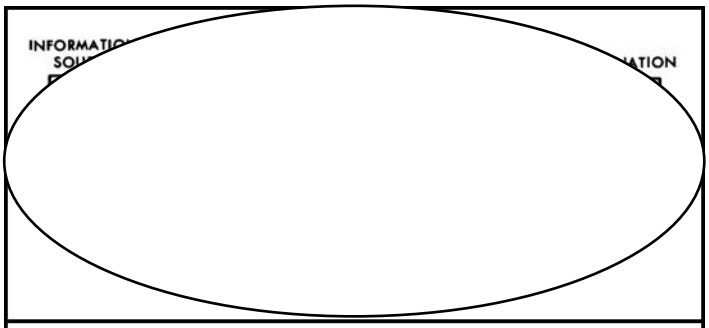

Fig.4. Schema of a communication system according to Shannon 1949/1960, p.34.

Since my purpose is just to explain the model underlying Shannon's law for measuring

[316] Shannon 1949/1960, pp.31-32.
[317] John Wilder Tukey (1915-2000) is credited also by Shannon with defining the bit.
[318] Shannon 1949/1960, p.34.

information, that is «[…] how much "choice" is involved in the selection of the event or how uncertain we are of the outcome»[319], and in turn this is of interest only insofar as it concerns the role of information in computational systems, I can confine my explanation to discrete noiseless systems and to *theorem 2* in Shannon's paper, the one in which the measure of information is stated. In fact, conceptually speaking information is always the same, even if of course *mathematically speaking* there are substantial differences, the main being remarked by Shannon himself: «there is one important difference between the continuous and the discrete entropies. In the discrete case the entropy measures in an *absolute* way the randomness of the chance variable. In the continuous case the measurement is *relative to the coordinate system*»[320]. Anyway, this variation does not make difference for the present purpose: I am interested in *what* is measured, more than *how*. For the same reason, nor I will discuss the demonstration of the theorem, in the cases of Nyquist and Hartley I did so because it was indispensable for understanding the underlying interpretation and compare it with Shannon's measure of information.

The role of probability in Shannon's measure of information consists of expressing uncertainty in the outcome of the process; as Shannon puts it «information is closely associated with uncertainty. The information I obtain when you say something to me corresponds to the amount of uncertainty I had, previous to your speaking, of what you were going to say. If I was certain of what you were going to say, I obtain no information by your saying it»[321].

Thus, consider the following. If a series of events with different probability (namely $p_1, \dots, p_n$) takes place, and they are listed exhaustively, then $\sum_{i=1}^{n} p_i = 1$ by definition, since they all together form a certain event. The measure of entropy H is defined as $-K \sum_{i=1}^{n} p_i \log_2 p_i$ [322], therefore the reader can see:

- if the events are all certain, namely there is only one event in the sum that has probability 1, then H=0, since in this case we have $H = 1 * \log_2 1 = 1 * 0 = 0$[323];
- information *is measured through a power of the number of options that can be selected at each step* (I assumed the binary system, so that the measure is in bit), *the power required to match the probability of a certain event*, scaled by a constant

[319] Shannon 1949/1960, p.49.
[320] Shannon 1949/1960, p.90. Italics in the text.
[321] Shannon 1950/1993, p.173.
[322] Shannon 1949/1960, p.50.
[323] Shannon 1950/1993, p.173.

that «amounts to the choice of a unity of measure»[324]. This becomes visible once that we set the multiplier to 1;

- multiplication of probability designates the case that independent events occur together; the sum of probabilities designates incompatible events, therefore *H* can be translated as: the information provided by a certain event p_i corresponds to the logarithmic relation of the probability that this event occurs *after all the independent, compatible events necessary for it to happen*, with the variety of choices available at each step of decision.

Put in this form, Shannon's law may seem quite abstract compared to that which it is commonly called a "message", but actually this description has a wide number of applications to many forms of communication. In his *Introductory Note on the General Setting of the Analytical Communication Studies*, Warren Weaver provides a framework for a clarification of the explanatory power of the new theory. First, it is necessary to distinguish *three levels of the communications problem*[325]:

A. *technical problem*: this is the main dimension of the measure of information[326], and the main concern of Von Neumann in his talk at the Hixon Symposium, as the main problem to face in order to ensure a response to the "effectiveness problem". The technical problem of communication is accuracy of transmission;

B. *semantic problem*: this is formulated as «how precisely do the transmitted symbols convey the desired meaning?». Thus, the problem is the efficacy of communication in linguistics sense: how much *clear* is the way in which the message is formulated from the standpoint of a possible receiver (in ordinary sense);

C. *effectiveness problem*: this is the question of how effectively the receiving of a transmission affects the behavior in a desired way. This was the main concern of Norbert Wiener in his *Cybernetics*.

That which makes interesting the theory for the others level according to Weaver's analysis is that «part of the significance of the new theory comes from the fact that levels B and C, above, can make use only of those signals accuracies which turn out to be possible when analyzed at level A. Thus any limitation discovered in the theory at level A necessarily applies to levels B and C. But a larger part of significance comes from the fact that analysis

[324] Shannon 1949/1960, p.50. In fact, *K* is omitted both in the subsequent lines and in Shannon 1950/1993, p.173.
[325] Weaver 1949/1960, p.4.
[326] This follows from the quotes of Shannon already provided. See also Weaver 1949/1960, p.6.

at level A discloses that this level overlaps the other levels more than one could expect»[327]. I want to focus on a particular intersection between levels B and C, which includes the semiotic conception of a sign I briefly mentioned in the footnote.

First, consider the following case. In a couple of his texts, Shannon provides the example of transmitting a message in English[328]. Instead of defining the probability of simple digits, it is possible to define the multiple layers of organization of the message in stochastic terms, as it follows: «the statistical structure can then be described by a set of transition probabilities $p_i(j)$, the probability that letter *i* is followed by the letter *j*. The indices *i* and *j* range all over the possible symbols. A second equivalent way of specifying the structure is to give the "digram" probabilities p_{ij}, i.e., the relative frequency of the digram *ij*»[329]. The same strategy can be applied to further levels of organization, as it is showed in pp.43-44 of Shannon's paper, and the result is that a text in proper English can be reconstructed with increasing precision, on the sole ground of statistical considerations. Moreover, notice that Shannon's theory *does not prescribe to specify the kind of signal used in communication*: this allows the theory and its consequences to be applied to every form of communication that respects the stochastic assumptions of the theory[330]. Thus, it is possible to use information theory for describing the transition between finite-state processes which implies compositional levels of organization and bounds for this transition, as it is the case of Turing's machine, for example.

Second, consider that which has been said about information. To sum up, information as it is defined in telecommunication science consists of a measure of the progression of resolution in a communication. Nyquist's work provided a form that is conceptually and mathematically suitable for treatment, Hartley provided some key concepts for interpreting the measure and revised some misleading assumptions in the work of his predecessor, then he reached the idea of information as resulting from series of choices and the sufficiency of the fundamental level of organization of the language applied. Information accounts for the message in terms of the stochastic bounds of the language used and the choices made up to a certain moment. Shannon found a suitable way for expressing the point, and for a full translation in mathematical terms of the conceptual intentions of his predecessors: even if Nyquist wanted to consider the structure of language, he could not translate this purpose

[327] Weaver 1949/1960, p.6.
[328] Shannon 1949/1960, pp.39-44; Shannon 1950/1993, pp.174-176.
[329] Shannon 1949/1960, p.41.
[330] There are many kinds of definitions of information that vary the mathematical constraints of the original theory, but they do not change Shannon's interpretation of the concept.

in mathematical terms; even if Hartley wanted to overcome the exponential progression of the law proposed by Nyquist and develop the problem in terms of choices, he could not define the relationship between the choices and the consequent, realistic progression of communication.

By putting together the necessity of considering the structure of the language and the aspect of choice, it is possible to understand the dynamic dimension of Shannon's theory: processes can be described in terms of probability of a transition from a certain alternative to another given the previous series of alternatives, and this transition mirrors the progressive organization of communication into meaningful strings of symbols. Thus, information theory is able to describe mathematically the transmission of meaning *as a transmission of the vehicle of meaning*. In the case of connectionist architectures, instead of communicating meaningful strings of symbols, information theory can be applied for describing the pattern of activation in order to measure the informativeness of the differential response of units. In some cases, it is also possible to describe them informatically in the same way classical devices are described, when a propositional calculus over them can be defined.

The possibility just described allows to think classical computing devices as acting according to an internal process of communication; communication of signals that stand for, but do not need for procedural purposes to send to, something different from themselves. Thus, they are themselves the content of communication. Nothing in Shannon's theory prevents this possibility. To sum up, it is possible to posit the flow of signals in classical computing devices as a process of *communication of well-formed formulas*, according to a twofold application of the concept of information:

1. the fact that transmission between internal components is structured according to the machine code version of high-level formal languages. Shannon's theory can describe in such a way any transmission linguistically structured by expressing statistically its syntactical and compositional bounds, as Shannon himself showed in his paper;
2. signals in computing machines are discrete choices in a binary system, operated by the process of self-control, as in the FDE cycle. Thus, as it is explained by Von Neumann and Weaver, signals in computing machines can be informatically analyzed as *stimuli with a certain probability of triggering a desired/planned response after a proper process of communication*, described by a formal calculus whose wff are that which is transmitted.

The picture that results is that classical computing devices manipulate, and act according to, an actual and proper formal language. In fact, through the proposed conception it is possible to state that the high-level formal language is no more a descriptive, functional construct, but the flow of signals as such: a series of events statistically connected in such a way to be compatible with the description provided by the program. The meaning of the symbolic strings depends on those structural features that warrant and implement a procedure coherent with the functional description correspondent to the formal language, as the user experience, but since in this new scenario the formal calculus is given to the machine too, the reliance on structural features assumes the meaning of the machine acting in accordance with *a pragmatical significance in semiotic sense* of the content communicated: signals as signs do not bring a content that is separated from them, they are themselves the content of communication instead, since the behavior itself is that which has to be transmitted; and it is the *procedural treatment that gives significance to the signals, their role within a practice*. In this case, signals are signs, but in a semiotic sense. Notice that, unlike the original semiotic signs, the pragmatical significance of those signals relies on *structural disposition to triggering certain actions*, instead of relying on *customary usages*. Here the ground of semantic content is *a structural disposition to triggering certain actions as consequences of certain symbols*[331]*, which can be informatically described*.

Now it should be clarified the connection between the three levels differentiated by Weaver. First, the problem of transmitting in a reliable way the strings of signals is not detachable from the problem of effectively affecting behavior. Given the explanations on the pragmatical features of interpretation in formal languages, it results that the internal process of communication between components is also a process of instantiating well-formed formulas of a given formal language, whose transmission – even if for structural/causal reasons – constitutes a transmission of *meaningful* signs. Thus, level A relates to level B, i.e. the latter consists of the former, since signs can also be meaning themselves, under the semiotic perspective I applied. Second, classical computers behave as computing machines in the sense their operations are defined by a configuration in Turing's sense, as I showed in §§1.2, 3-4 of this chapter, namely the behavior is triggered by the combination of a certain state and a certain input sign. Thus, levels A and C are bounded, and since signs and meanings are there indistinguishable, then B affects C by the

[331] Notice that this is a very sophisticated, non-linear stimulus-response model. I am not going to push on this, since this is an objection that it is not philologically accurate, nor it is really informative for the problem of what a representation is.

former being the communication process itself. It follows from this that levels B and C are intimately connected within the framework of information, and that, in classical computing devices, signals here are *informative* in the sense they are related to each other in a meaningful way and as such they necessarily trigger planned behaviors. At the moment, I confine the discussion to classical computational devices. The results here proposed will be extended to the connectionist case in the next chapter.

Now I start from this set of premises, therefore I ascribe representations to computing devices under the assumption that they manipulate symbols qua symbols, since they have a pragmatical approach to their interpretation; then I remove the postulate of the need of superimposing a content over another content for interpretation to be given, and I draw the consequences of this hypothesis. I am going to show that there is a flaw in every behavioral view of semantics as it is this, one that had already been discussed in philosophy in the context of Hilbert's project of finitist arithmetic, his debate with Intuitionism, Frege's view on the topic and Cassirer's comment to this debate.

§7. The Problem of Sign in Information Theory Is the Same as in Finitist Arithmetic

7.1. The Formal Treatment of Language: Frege on Evidence and Demonstrations

Traditionally, the role of sign has been instrumental regarding knowledge, compared to the role of declarative contents: the first was just a way of annotating the former, it played no role in the formation of content. The project of a formal calculus appeared only with the project of a *mathesis universalis*, namely in the Seventeenth century, and one of its remarkable revolutions is that it began to change this background assumption, which will be explicitly discussed only in the Nineteenth century, by Frege and Hilbert, and furtherly in the Twentieth century by authors as Cassirer, Derrida and Barthes. As I anticipated, I will discuss only the debate between Frege and Hilbert, since they are closer to the question of computation, and the comment of the topic of the role of sign in forming thought-contents by Cassirer.

There is a nexus that unifies both Frege's conception of formal calculus and Hilbert's project of finitist arithmetic and his thoughts on formal calculus: *the problem of verifiability of our intellectual operations*. Both the authors were interested in formal calculus as a mean for a reliable foundation of mathematics and geometry on logic, especially for dealing with the new mathematic of infinite on a firm ground. The two authors approach the question with the same interest but different perspectives. On the one hand, Frege's attitude and its interest in *Ideography* is still the one expressed in the Seventeenth century of providing a robust ground for intellectual operations by defining a

reliable process of deduction, one that leaves unvaried the transmission of truth from basic, self-sufficient elements up to more complex truths. Thus, he could not regard a pure symbolic operation as a solution, since for him a formal calculus must guide – not constitute – the discovery of self-evident truth from which deriving more complex discoveries on autonomous and independent objects. On the other hand, Hilbert's project of Finitism stems from the necessity of finding a reliable foundation for the incoming mathematics of infinite, but he regarded the issue of truth and foundation as a problem of consistency, one that a purely formal procedure can handle by itself[332].

I begin my exposition from Frege, who comes first in order of time. In *The Foundations of Arithmetic*, it is said: «to minimize these drawbacks, I invented my concept writing. It is designed to produce expressions which are shorter and easier to take in, and to be operated like a calculus by means of a small number of standard moves, so that no step is permitted which does not conform to the rules which are laid down once and for all»[333]. The "drawbacks" are the excessive length of demonstrations resulting from the necessity of developing them step-by-step for jumps to be avoided in the reasoning, in which mistakes may take place; moreover, natural language impairs two desiderata of logical method: individuating a limited number of inferences that are valid for all the purposes, and the possibility of embracing them with a single glance. Thus, here the reader may see that Frege's interest in an ideography lies in the *evidence of a process of reasoning*. For according to this claim, over than constructing a rigorous language for handling our intellectual activity[334], Frege's point in *Ideography* is to make evident the steps of reasoning by making them suitable to be *considered at once*[335].

---

[332] Resnik 1947, p.395: «Both problems (mathematical truth and mathematical existence) had been sharply accentuated by the discovery of non-Euclidean geometries and the introduction of imaginary and complex numbers. Frege approached these problems via reductionism. The questionable mathematical entities were to be defined in terms of entities of evident existence – one was not simply to postulate their existence. On the other hand, the truth of mathematical theorems was to be reduced via an appropriate axiomatization and set of definitions to self-evident or obvious mathematical truths. Hilbert, by contrast, placed no importance upon the reduction of one theory to another. Indeed, the closing paragraph of "Über den Zahlbegriff" implies that reductionism is an unfruitful approach to the problems of truth and existence as it concerns the theory of real numbers». Resnik 1947, p.395 also reports a quote from Hilbert's correspondence with Frege in which he says that «[…] if arbitrarily postulated axioms do not contradict each other with their collective consequences, then they are true and the things defined by means of the axioms exist».

[333] Frege 1884/1960, p.103.

[334] See on this Frege 1878/1967, p.7: «if it is one of the tasks of philosophy to break the domination of the word over the human spirit by laying bare misconceptions that through the use of language often almost unavoidably arise concerning the relations between concepts […] then my ideography, further developed for these purposes, can become a useful tool for the philosopher»; Sluga 1980/1999, pp.65-71.

[335] Frege 1884/1960, pp.102-103. See especially the passage at p.103: «[…] to isolate a set of modes of inferences which is both sufficient to cope and easy to take in at a glance». In the Italian translation by Ludovico Geymonat and Corrado Mangione, the same passage reports that "a set of inferences, that can be embraced by our intellect" (my translation, original as follows: «[…] un quadro preciso di forme deduttive, che risulti immediatamente afferrabile dal nostro intelletto»). The original text in German is: «Dazu kommt,

The same topic is discussed by Descartes in the sixteenth of the *Regulae ad Directionem Ingenii*: «but, as for the things which, even if they are necessary for the conclusion, do not require the present attention of mind, it is better to represent them by very concise symbols rather than by complete figures – for thus will it be impossible for the memory to be misled, nor yet will thought be distracted in the interim by having to retain these things while it is involved in deducing other matters»[336]. The reader may appreciate that the point is in both cases to make demonstrations tangible and verifiable with a simple and immediate act of understanding. In fact, in both cases the necessity of a symbolic procedure is due to the discursive nature of human understanding, namely its property of considering things one by one, in a series. By making reasoning tangible and permanent through writing, intellect can consider with a simple act (seeing) both the elements that form a reasoning and their connections[337]. This line of thought emphasizes the importance of making each step of rational procedures evident.

As a coherent rationalist, Descartes conceives evidence as a non-psychological act that depends on intellect as opposed to sense, imagination and memory: intellectual "intuition", in the sense he understands it, is the *cause* of the feeling of grasping a truth clearly. In fact, Descartes describes intuition as follows: «by "intuition" I understand neither the fluctuating testimony of the senses nor the deceptive judgment of an imagination which composes things badly, but rather the conceptual act of the pure and attentive mind, a conceptual act so easy and so distinct that no doubt whatsoever could remain about what we are understanding. Alternatively, it amounts to the same thing to say that by 'intuition' I understand the indubitable conceptual act of the pure and attentive mind, which conceptual act springs from the light of reason alone. Because this act is simpler, it is more certain than deduction, which, however, as we have noted above, a human being also cannot perform wrongly»[338]. Of course, defining such a thing as intuition in these terms was one of the main difficulties in Descartes' thought.

Frege has a similar view in mind, but is not concerned with intuition as a faculty, he cares more of the being evident and the role of evidence in demonstrations as a process. This is more a difference on emphasis than on content, since Frege also consider concepts and theoretical entities as something that need to be discovered, therefore understood with a clarity that makes his view compatible with Descartes' gnoseology, if not even

dass die übergrosse Mannichfaltigkeit der in der Sprache ausgeprägten logischen Formen es erschwert, einen Kreis von Schlussweisen abzugrenzen, der für alle Fälle genügt und leicht zu übersehen ist».

[336] Descartes 1619/1998, p.197.

[337] For the same things in Descartes, see Descartes 1619/1998, pp.136-137.

[338] Descartes 1619/1998, p.79.

presupposing a similar concept of intuition in the background, perhaps not applied to whole entities but to basic assertions on them[339]. Frege distinguishes between how a certain judgment is formulated and how the same judgment is demonstrated[340], and he maintains that the purpose of demonstrations consists of showing the dependence of truths one on another[341], therefore there are basic laws that must ground all the others[342]. According to Frege, at least until Russell had shown his famous paradox in set theory, these must be the truth of logic, on which even arithmetical propositions are grounded[343]. How such self-contained grounds are recognized? An indirect clue is provided by the alternatives given about the demonstration of the proper concept of number for arithmetic: «if we now try to meet this demand (my note: not doing any jump in the chain of deduction) we very soon come to propositions which *cannot be proved so long as we do not succeed in analyzing concepts which occur in them into simpler concepts or in reducing them to something of greater generality*»[344]. Thus, according to this passage, a demonstration ends when something that has the following features is found:

- it cannot be proved;
- it cannot be made simpler;
- it cannot be reduced to more general elements, i.e. it cannot be derived from other elements, since we are in the context of deduction.

In Frege's production, that which fulfills these requirements are the logical operators: in *Ideography* they are not demonstrated, they are described by providing their truth-tables, under the reason that expressing them is impossible within the system, since they constitute its basis[345], and as such its conditions of possibility. If this is the case, then Frege regards as evident only the most fundamental connections and their directly related rules of inference – as Descartes shows to think too, since he grounds his whole system on a material implication, the famous *cogito*. Moreover, according to the explanation about the utility of the *Ideography*, evidence of demonstrations becomes something close to Descartes' eleventh rule: «if, after we have intuited a number of simple propositions, we want to draw some other conclusion from them, then it is useful to run through them in a continuous and completely uninterrupted movement of thought, to reflect on their mutual

[339] See on a similar view Burge 1998.
[340] Frege 1884/1960, p.3.
[341] Frege 1884/1960, p.3.
[342] Frege 1884/1960, pp.4-5.
[343] Particularly clear on this is Frege 1885/1991, pp.112-113.
[344] Frege 1884/1960, p.5. Italics added.
[345] Frege 1878/1967, p.28.

relations, and, as far as possible, distinctly to conceive several of them together – for thus does our knowledge also become much more certain, and the capacity of the natural intelligence is increased as much as possible»[346]. In fact, Frege declared that some of the importance of *Ideography* lies in making evident complex inferential chains by making them visible and liable to be embraced at a single glance, and those in turn are useful for showing the truth of remote assertions by showing their connection with more basic and self-evident truths.

To sum up, Frege and Descartes share the idea that some kind of evidence lies at the basis of knowledge, and that the utility of writing reasoning through a symbolical language lies in *extending this evidence to inferences*, so that mistakes are prevented by making reasoning present to itself, visible, concrete, as in the case of the most basic truths. Moreover, as Frege conceives *Ideography* as a mean for developing reasoning in such a way that it is guarded by the threat of subjective intuition[347], in the same way Descartes is concerned with the mistakes that come from memory and imagination during the process of reasoning[348]: it is possible to say that Frege provides his *Ideography* also as an answer to the Seventeenth-century need of saving reasoning from the mistakes that the other faculties may add to the activity of the understanding. There is also a further common ground between Frege and the Seventeenth-century epistemology formulated by Descartes: both admit as a reliable source of knowledge only intuition and deduction. Descartes explicitly does so[349], Frege does with his conception of deduction and his foundationalist perspective, and in both case "intuition" must be understood as an intellectual act, not as a psychological state. In the end, both Frege and Descartes regards as self-evident only the *rules of logic*: neither Frege nor Descartes consider objects – intelligible and surely not empirical – as self-evident, since in this case Frege would regard as useless an enquiry on the concept of number or on a logic of discovery, and Descartes would not engage radical skepticism.

In accordance with such a perspective, Frege is straightforward regarding the self-sufficiency of a purely formal calculus regarding reasoning. In general, in his *Fundamentals of Arithmetic* he states that «it is possible of course to operate with figures mechanically, just as it is possible to speak as a parrot: but that hardly deserves the name

[346] Descartes 1619/1998, p.135. The *Third Rule* also states something similar, and it also remarks the opposition between certainty as deriving from intellectual act of intuition (comprehension) and certainty as coming from sense, memory or imagination.
[347] See again Frege 1884/1960, p.102.
[348] Descartes 1619/1998, p.121 and following.
[349] Descartes 1619/1998, p.79.

of thought. It only becomes possible at all after the mathematical notation has, as a result of a genuine thought, been so developed that it does the thinking for us, so to speak»[350]. Thus, Frege admits the hypothesis – such it was at his days – that a purely mechanical operating can carry on results on its own and spare the effort of the procedure, but at the same time this is something that mimic thought, as the apparently smart speaking of a parrot. In the quote, Frege alludes to "the result of a genuine thought": this means that some kind of thought must operate *before* a formal writing can be produced. That which must operate is reasoning itself: *Ideography* is supposed to represent reasoning, therefore a reasoning must be specified in advance, before it can even be translated into signs. In fact, it has already been mentioned that it is impossible to demonstrate logical operators because they are that which makes possible the whole *Ideography*: their consequences can be described by *Ideography*, for example through the truth-tables, but they cannot be demonstrated, in the sense "justified by something different", nor formed independently by it.

7.2. Hilbert's Finitism

The first theses by Hilbert regarding a development of a self-sufficient formal calculus as related to foundational issues can be found in his paper *The New Grounding of Mathematics. First Report* in 1922.

Hilbert rejects both Frege and Dedekind's realist approach to mathematics, which is a non-viable position for him, even if the two authors are credited with beginning the modern critique of mathematical analysis[351]. In contrast to this, Hilbert proposes a construction through his axiomatic method, grounded on the intuition of objects. He explains that «[…] as a precondition for the application of logical inferences and for the activation of logical operations, something must already be given in representation: certain extra-logical discrete objects, which exist intuitively as immediate experience before all thought. If logical inference has to be certain, then these objects must be capable of being surveyed in all their parts, and their presentation, their difference, their succession (like the objects themselves) must exist for us immediately, intuitively, as something that cannot be reduced to something else»[352]. The reader may see that also in this case the problem is still the one shared by Frege and Descartes: the necessity for thought of becoming concrete, being subtracted by abstraction and subjectivity in order to become visible, prone to be surveyed

[350] Frege 1884/1960, p.XVI.
[351] Hilbert 1922/2007, p.1121.
[352] Hilbert 1922/2007, p.1121.

at a glance, and therefore viable to be checked under the light of evidence.

Hilbert's position, as I anticipated, differs in the solution. These "objects" are actual object: Hilbert uses the word "Gegenstände", which designates "that which stands in front of", thus something referred usually to concrete objects. In fact, Hilbert explicitly posits concrete signs as these pre-logical items, on which logical operation should be grounded. According to Hilbert, «because I take this standpoint, the objects of number theory are for me – in direct contrast to Frege and Dedekind – the signs themselves, whose shape can be generally and certainly recognized by us – independently of space and time, of the special condition of the sign, and of insignificant differences in the finished product»[353]. The preliminary condition of logical understanding, thus, is no more the clarity of intellectual intuition, but the evidence of sensible intuition, the manifest presence as such.

Notice the difference: according to Frege, as well as to Descartes, the presence required was the self-evidence of mental understanding to itself, both regarding inferential steps and objects elucidated by them; whereas here we meet sensible presence as the ground of logical operation and logical objects – that which was instrumental and vicarious, now it is the content itself. In fact, the idea that signs are the *constituent* and not the *notation* of the theoretical objects of mathematics and geometry has to be taken that radically, since Hilbert in a subsequent passage declares: «These number signs (Zahlzeichen), which *are* numbers and which *completely make up the numbers*, are themselves the object of our consideration, but otherwise they have no *meaning* (*Beudeutung*) of any sort»[354]. If this is the case, proofs are about signs too, no more about logical laws as abstract entities, since there is nothing beyond signs, and this holds for the objects as well as for those laws of thought that Frege claimed can be just described. Hilbert indeed writes that «we simply have concrete signs as objects, we operate with them, and we make contentual [inhaltliche] statements about them. […] I should like to stress that this proof is merely a procedure that rests on the construction and deconstruction of number-signs […]»[355].

Of course, such a procedural and empirical view is problematic, and according to Patton 2014, Aloys Müller in his *Űber Zahlen als Zeichen* objected the following: «one might recall Hilbert's remark that for him, as opposed to for Frege and for Dedekind, the objects of number theory are "the signs themselves." Müller's objection boils down to the following. If signs are concrete, intuitable objects and are "without meaning," then they

[353] Hilbert 1922/2007, p.1121.
[354] Hilbert 1922/2007, p.1122.
[355] Hilbert 1922/2007, p.1123.

must function as signs in virtue of their concrete, intuitable properties, that is, in virtue of their observable form or shape. But if those shapes don't indicate other objects, they no longer function as signs at all, only as brute shapes or figures. But Hilbert says the signs are the numbers. In that case, the foundation of number theory on Hilbert's view is the manipulation of shapes that do not indicate any further object or phenomenon: "mere" "signs without meaning"»[356]. As Patton remarks, Müller perceives Hilbert's account as problematic because in his view there is a contradiction between being sign and having a shape without meaning[357].

Hilbert is aware of the objection of course, and in fact his theory includes a response to the problems of reference and interpretation, which is expressed in Hilbert's definition of his method as *axiomatic* and *contentual*. In a passage, he contrasts statements that can be constructed by progressive replacements[358] to procedures that implies generalizations, for example the principle of induction, and he says that it is impossible to construct statements concerning wide generalizations by composition of finite figures; for them, actual formulas must be generated[359]. The solution offered is: «we can achieve an analogous point of view if we move to a higher level of contemplation, from which the axioms, formulae, and proofs of the mathematical theory are themselves objects of a contentual investigation. But for this purpose the usual contentual ideas of the mathematical theory must be replaced by formulae and rules, and imitated by formalism»[360]. In short, Hilbert's point is to follow the example of Boole's or Frege's formal calculus, but from a different perspective. The point is no more annotating reasoning in order to favor intellectual glance, but including the rules and the formulas of reasoning into a construct of signs together with their arguments, so that rules, formulas and their application become as sensible and intuitable in Hilbert's sense – that signs are the sole and only constitutive content of logico-mathematical thought – as their arithmetical counterparts, therefore it becomes possible operating on them in the same way as on pure shapes. Thus, *"contentual" here is referred to "having a content that can be received through senses"*. If this was the whole story, then nothing would be solved, but the point is that which lies behind the *contentual symbol*.

The concept of "being provable" proposed is the one that follows from these premises: «a

[356] Patton 2014, p.192.
[357] Patton 2014, p.192.
[358] I mean statements as $a + b = b + a$. In this case, it is possible simply to operate a replacement according to pre-given definitions. The same holds for constructing numbers as sets of progressively added units, as in the statement $1 + 1 + 1 = 3$.
[359] Hilbert 1922/2007, pp.1122-1123.
[360] Hilbert 1922/2007, pp.1123-1124.

formula is said to be *provable* if an axiom or results from an axiom by substitution or is the end-formula of a proof or results from such an end-formula by substitution. Thus the concept "provable" is to be understood relative to the underlying axiom-system. […] in our present investigation, proof itself is something concrete and displayable; the contentual reflections follow the proofs themselves»[361]. This point is clarified in a previous work, *Axiomatic Thought* (1918). Hilbert maintains that his method is *axiomatic* in the sense that few *clear* (not self-evident as manifestly true) propositions are *sufficient to construct* the subsequent concepts, through which the whole system can be developed simply by deriving the consequences of their (of concepts) composition and analysis through a logical method, and this has an explanatory value because of a primitive correspondence between that which has to be explained and the order that those systems of axioms, concepts and their derivations bring by constructing logically that which they map[362]. Hilbert calls this method "axiomatic" because it is the same thing that, according to him, Euclid did in his geometry. Thus, if the relation of mapping, on which truth depends, is a presupposition that lies in the axioms and it is preserved with the preservation of truth in the process of deriving complex statements, then all that a scientific system must warrant is (1) *consistency* and (2) *clarity on dependence (and independence) of propositions*[363]. This is why Hilbert regarded the question of foundation of arithmetic on logic as dependent on the above-mentioned *Entscheidungsproblem*, whose solution was Turing's and Gödel's aim[364]: if consistency of axioms and consistency (as truth-preserving) of derivation is enough for achieving truth, then a reflection about proving in general and specifically proving consistency is the most important piece of the theory, and it is that which Hilbert's Finitism is – the idea that a proof of consistency must be achieved by discrete ideas and clear operations within a discrete number of clear passages[365]

In the light of this account, Hilbert's theory of proof is contentual in the sense symbols are contentual contents, namely sensible contents; Hilbert's system is *contentual* also in a further sense, over than this one: not only in the sense that a content must be provided to

[361] Hilbert 1922/2007, p.1127.
[362] Hilbert 1918/2007, pp.1107-1109. See in particular p.1108: «if we consider a particular theory more closely, we always see that a few distinguished propositions of the field of knowledge (my note: the set of objects mapped by the concepts that form the theory, whose logical relations map the set of facts, p.1107) underlie the construction of the framework of concepts». This means that these few propositions, which are the axioms (p.1108), *are sufficient to describe the objects and to construct the concepts, whose combination by logical connections make them sufficient to describe also the facts*, as it is said in the quote.
[363] Hilbert 1918/2007, p.1109.
[364] And this is in fact stated at Hilbert 1918/2007, p.1113.
[365] Cfr. Hilbert 1918/2007, p.1115. For Finitism in Hilbert's thought as a theory of proof implying the concept of effective procedure, see Sieg 2009 and Patton 2014.

sensible intuition, but also in the sense that there is a basis of facts that are symbolically mapped and constitute the underlying reference of the axioms, i.e. *their content*, which can safely be updated to every need consistent with achieving this relation of mapping as well with achieving consistency[366], and the subsequent manipulation of symbols *qua symbols* allows to extend the validity of the basic contents to further and more complex statements, which in turn constitute a further articulation of the basis of facts. If this is the case, Hilbert's solution for granting the mathematics of infinite as well as general procedure consists of *extending the basic content to dependent propositions by deriving the non-intuitable from the intuitable – both in being a discrete symbol and in being a self-contained and discrete object or fact – content by a valid procedure of derivation*, derivation that in the case of logic/mathematics is pure symbolic substitution, as I am going to show discussing *On the Infinite* (1926).

A piece of the answer is in fact still missing. In *Axiomatic Thought*, I have reviewed Hilbert talking about objects and fact, but in *The New Grounding of Mathematics*, I showed Hilbert rejecting Frege's and Dedekind's realistic stance on mathematical entities. Therefore, there is no room left for such a thing as "mathematical objects" or "mathematical facts", as well for the correspondent items in geometry, in Hilbert's system. The answer to this point is given in *On the Infinite* (1926). Since Hilbert's Finitism recommends just consistency, specifying hierarchical relations and validity in deriving consequences for propositions as all and only conditions for truth preserving *starting from axioms*, then all consistent sets of axioms are true and *demonstrative of existential claim* regarding objects whose truth depend ultimately on self-mapping. In fact, Hilbert replies to Frege that consistency of properties is enough for ascribing existence[367], and that existential claims are equivalent to "at least one thing fulfills this set of properties", namely to the satisfaction of a set of conditions[368]. For the sake of accuracy, Hilbert and Barnays will distinguish the idea that a set of axioms maps a pre-given (fixed) set of objects from the idea that objects are introduced as members of a class, and will call the former *axiomatic method*, whereas the latter *constructive or genetic method*[369]. Thus, even if both methods are grounded on axioms, Hilbert must be considered a constructivist in mathematics and geometry, whereas he is more properly defined an "axiomatist" regarding natural sciences. *From his point of*

[366] Cfr. the case of Zermelo at Hilbert 1918/2007, p.1112, who constructed the definition of "set" in order to avoid the paradox of the set of all sets stated by Cantor.
[367] See the first letter of Hilbert to Frege. I read it through Resnik 1974, p.395.
[368] Hilbert 1925/1967, pp.377-378.
[369] The relevant passage is in Barnays and Hilbert (1934) *Grundlagen der Mathematik*, vol 1. I read the quote in Patton 2014, p.199.

view, Hilbert is not a formalist, as his detractors claim, for example Aloys Müller. For from his standpoint, Hilbert ascribes a content to his signs, in the sense that they *presuppose* one, and until this presupposition can be granted – and it can always be through a simple conventional definition – Hilbert's signs receives a meaning indirectly; the point is that this meaning plays no role in the procedure of demonstration, neither in the axiomatic method nor in the constructive one. In the case of the mathematics of infinite, all that is demanded consists of defining the rule of substitutions between *contentual items and general formulas and rules of inferences*. It amounts to defining what sets of symbols can take certain material places in pre-defined formulas and rules that are contentual (symbolic) as well[370]. Frege did something similar when he says that places in functions are reserved either to names or concepts, and this will be relevant in their debate[371].

7.3. Cassirer on The Problem of Defining the Fundamental Entities and Relations Within Formal Systems

Cassirer dedicates several passages of his *Philosophy of Symbolic Forms* to the role of symbols in mathematical sciences. It must be considered that in this work he had access only to texts up to 1927, the year in which he declares to finish his book.

Cassirer reads Hilbert as the promotor of a new program, centered on the shift that I attempted to elucidate[372] from demonstrating as a matter of intellectual acquisition under the principle of clarity, understood as evident understanding through symbolical annotation, to demonstrating as a matter of constructing demonstrative content through symbolic production that allows the guidance of sensible intuition. In Cassirer's words: «the process of verification is shifted from the sphere of content to that of symbolic thinking. As precondition for the use of logical inferences and for the practice of logical operations, certain sensuous and intuitive characters must always be given to us. It is in them that our thinking first gains a sure guiding thread, which it must follow if it wishes to

[370] Hilbert 1925/1967, pp.379-381.

[371] On the Frege-Hilbert debate, see Resnik 1974 and Blanchette 2018. Frege does not object to Hilbert his ontological conception of mathematics, but his use of definitions as axioms and of axioms as bricks for constructing mathematical entities. Frege regards this strategy as a categorial mistakes between concepts of concepts, concepts of objects and objects. His criticisms are expressed in two series of papers called *Foundations of Geometry, First* and *Second series*, respectively 1903 and 1906.

[372] For the sake of intellectual honesty: first I read Cassirer, then I read Hilbert. Here I mean that I attempted to elucidate the transition by comparing the writings by Frege and Hilbert, not that I formulated independently a reading consistent with Cassirer's, nor anyway my reading of Hilbert is based only on the indications of Cassirer, as the reader can see from the texts I examine. About this, Cassirer also compares the question of making reasoning visible through symbolic production in the Seventeenth century tradition to Boole's idea of formal calculus, but he does this by examining Leibniz, who in fact is the one who most developed the topic of formal language. Descartes is mentioned only at the end of the section I am going to survey, regarding the question of evidence.

remain free from error»[373] and «here the position and function of intuition are entirely different than in the intuitionistic foundation of mathematics. Its role is not active, as in intuitionism, but passive – it is a kind of datum, not a mode of giving»[374]. Thus, according to Cassirer, the point in Hilbert's theory is making visible thought so that both it can orientate in the questions of pure thinking and it can be self-supervising, but not as in other authors by becoming concrete and remaining at the same time a content that is denotated, instead by becoming *only* symbolic production; for this reason, symbolic capacities and symbolic materials become the conditions of possibility of consistent thought, at least in scientific thinking, and it becomes the only object of the scientific enquiry. In this sense Cassirer says that sign is "a kind of datum, not a mode of giving": there is nothing that is given through the sign, the sign is that which is given; moreover, symbolic production forms content, so that the internal economy of the system of signs shapes both content and how it is given.

This claim must not be understood as if the material sign had to be the object of sciences insofar as it is a concrete singular, that can be indicated. If this would be the case, opponents of Hilbert would be right in claiming that Hilbert's system of signs is just the production of a sensible game[375]. I have shown that Hilbert's position is indeed sharper and more complex, and that its core consists of a form of constructivism regarding mathematical and geometrical entities. Cassirer argues that Hilbert's constructivism has its own necessity: whenever the sharp distinction is drawn between the means of mathematical expression and the objects of mathematical enquiry, as Weyl does, together with this author we are compelled to claim that knowledge beyond sensibility is a form of faith, since humans go beyond the reach of senses (which is not the reach of experience) only through theoretical and symbolic reasoning[376]. Thus, either one accepts the dichotomy between sensibility and intellect, or an attempt is made of recollecting them within an order that includes both and is grounded on their cooperation, as Cassirer does by looking at signs and symbolic production as historical instances of a transcendental activity of making the world as the set of all objects[377].

According to Cassirer, Hilbert's theory has to be understood within a general tendency that his enquiry has highlighted, namely the fact that symbolic production in human activities – speaking, mythical and artistic narration, knowing – overcomes the classical distinction

[373] Cassirer 1929/1957, p.379.
[374] Cassirer 1929/1957, p.380.
[375] Cassirer 1927/1956, pp.380-381.
[376] Cassirer 1927/1956, pp.382-383.
[377] Cassirer 1927/1956, pp.383-385.

between vehicle and content by forming objects "in his own image, in his own likeness", i.e. each of them is a kind of exercise of the same formal function of forming objects according to a certain economy, that is transferred into a certain mode of aesthetic production and presentation. Cassirer expresses this perspective regarding mathematics in this way: «if we now apply this universal insight to the world of mathematics, we find ourselves, here too, raised above the alternative of dissolving the symbols of mathematics into mere signs, into intuitive figures without significance, or of endowing them with a transcendent significance which only metaphysical or religious faith can reach. For in either case we should be missing their proper meaning. This meaning does not consist in what they "are" in themselves, nor in something that they copy, but in a specific trend of ideal formation - not in an outward object toward which they aim, but in a determinate mode of objectivization»[378].

According to Cassirer, from the standpoint of a function of objectification, Hilbert's point of view has the merit of showing the importance of symbolic reasoning as finally liberated from the problem of transcendent meaning. Hilbert is credited with solving the Cartesian problem of subtracting discursivity from having to be grounded on memory of successive passages, so that it can be instead become evident through sensible intuition[379]. According to Cassirer, Hilbert does that by overcoming the whole perspective, since he shows that only through symbolic competence these concepts can be *formed* and not simply discovered – a position that Cassirer regards as essentially entangled into a false dichotomy that needs to be defused. Otherwise stated, the merit of Hilbert lies in showing that mathematics *deepens* his concept-formation through symbolic production as a moment of construction self-contained[380].

Hilbert is instead wrong in considering the process of symbolic production as self-sufficient regarding its own definition, instead of self-contained as heuristic method. Cassirer thinks that mathematical entities and relations are *ordering relations*[381], therefore they can be applied, but not produced within the process that applies them: their mode of

[378] Cassirer 1927/1956, p.383.
[379] Cassirer 1927/1956, pp.387-389.
[380] Cassirer 1927/1956, p.389: «on the whole we may say that intuitive thinking provides the foundation of the mathematical edifice, while symbolic thinking builds it up».
[381] Cassirer 1927/1956, p.384: «But there is a definite gradation from the logical to the empirical, from the pure form of thought to the object of experience, and here the mathematical appears as an indispensable transition. In contrast to the logical object, the mathematical object discloses an abundance of new, concrete determinations; for to the universal form of postulation, of differentiation, of relation it adds a definite mode of postulation, the specific mode of postulation and ordering that is represented in the system of numbers and in the natural numerical series. But on the other hand, this new mode proves to be the indispensable preparation and presupposition for the achievement of an order in the world of perception and hence of that object which we call the object of "nature"».

organization must be already available in order to be applied, and they are applied in concepts as space, number or time. By devoiding mathematical formalisms of *any* reference and claiming that this is a self-sufficient way of doing mathematics, both in the theory of proof and in the construction of mathematical entities, Hilbert fails in understanding that all his constructions – rules of substitution, rules of inferences, numerals – all presupposes a pre-formed mathematical function that is that which makes these construction both comprehensible, possible and meaningful[382]. Thus, if devoided of any reference to them, they not only lose any accountability and rationality, but they result even "miracles", meaning that they are impossible constructions once that they are detached from the conditions of possibility that they make intuitable.

The outcome just stated is due to the fact that not every formula can collapse into a concept-writing in general: principles and rules with the role of principles in the proper sense will always be impossible to express within any formal system without presupposing themselves, since they constitute its condition of possibility. This is the same case of logical operators in Frege's *Ideography*, and in fact Cassirer's criticism is a sufficient premise for attributing a lack of self-interpretability to every formal system, as those presupposed by Computationalism or that one constructed by Frege, as I am going to show in the next subparagraph.

7.4. The Same Problem Applies to any Behavioral View of Interpretation, as the One Implied by Information in Computing Devices

Now it is possible to resume the question of signals in computing devices as semiotic signs, so defined through information theory. In §5, point 4 of the first list, I discussed the question whether symbolic discrimination may imply symbolic competence. The debate on Frege's and Hilbert's positions on a purely formal calculus as an independent and autonomous activity and the review on Hilbert by Cassirer helps in shedding some light over that topic.

Hilbert's discussion of the finitary approach shows another aspect of formal languages: the

[382] Cassirer 1927/1956, pp.386-387: «for Hilbert would not have been able to build up and enlarge his system of signs if he had not based it on the primordial concepts of order and sequence. Even when they are taken as mere signs, Hilbert's numbers are always positional signs: they are provided with a definite "index" which makes intelligible the mode of their sequence. Even if we regard the individual signs as nothing more than intuitively given, extra-logical discrete objects, these objects in their totality never stand simply side by side as independent elements, but possess a determinate articulation. If we start from 0 as the initial sign, we arrive by a definite progression at a "next" sign 0 ' and thence at 0", etc. Ultimately this means only that if the individual numbers are to be securely differentiated, they must be kept apart in a definite order, and this keeping apart is in itself fundamentally an enumeration, a term used in the sense of "content." The strokes that we use to separate 0 from 0', 0" from 0"' etc. function already as numbers in the sense of a purely ordinal derivation of the numerical concept».

content they carry may be the signs themselves, on their own, as pure sensible items offered to perception. This is of interest for the research on the problem of representation. According to the analysis in §6, since information theory defines the pragmatical significance of the signals within computing devices, these signals can be considered semiotic signs, and now we may see that they are symbols in Hilbert's sense: their manipulation follows rules that are in turn part of, and expressed within, the same formalism that defines their significance as signs, since they are manipulated according to the rules defined through these formalism and within their particular cases (programs). In virtue of their structure and their purely formal manipulation, signals that carry information seem to be signs without the division in sign-vehicle and content to be demanded. Moreover, the case of Hilbert's finitary approach shows that this kind of signs may also become a sufficient mean for developing knowledge on fields typically considered as rational enquiries, whenever the formal languages at stake presuppose consistency and inferential validity as sufficient means for truth. Moreover, Cassirer argues that this kind of procedure overcome the problem of representation as interpretation – which I postulated as presupposing the duplication and superimposition of a content on another – in this: symbolic production may become the mean of *formation and not simple annotation* of contents that are otherwise unreachable.

It seems to me that this is the only way of making claims on signals in computing devices having semantic properties consistent with material features and concrete role these signals have in computing devices. This is not important anyway: it is enough that the conceptualization proposed is appliable.

Thus, if signals manipulated by information-processing procedures for implementing computational operations are signs at all, then they are signs in a semiotic sense, which make them compatible with Hilbert's finitist approach to computation. Indeed, this is also consistent with the connection between the idea of effective procedure and the question of computing in computer science, as it has been showed in previous paragraphs. Moreover, it seems that Hilbert's finitary approach offers a viable solution for conceiving computational devices as operating genuine symbolic manipulations, over than a ground for defending the ascription of rationality to them in their operations.

Even in this case, despite one is not compelled to the alternative between a realistic stance as Frege, which regards production of sign as an annotation and an instrumental mean for intellectual evidence, and a pure formalist stance, namely one that regards formal language as a pure game of figures, according to Cassirer the flaw remains that these systems are

self-sufficient devices for developing proofs, but not self-contained regarding their own foundation and accountability. The reason for this is that (1), if they claim excessive independence, formal languages lose the reference to their epistemology and ontology, since when they are detached from reference to their foundation, they cannot preserve the mapping with that which they put in condition to deepen, therefore they become formalistic instead of constructivist. A second reason for the impossibility of considering formal languages self-contained is that (2) formal languages have their more fundamental rules and objects as a permanent beyond, a presupposition that they cannot reach but on which their possibility depends, therefore they become unaccountable on their own means. Cassirer shows that, paradoxically enough, the reason for this issue depends on the ground of their impressive explanatory and heuristic power: by expressing rules in finitist terms, namely in terms of signs and manipulation through an effective procedure of derivation as symbolic substitution and composition, formal languages make thinking visible, and make available both an astonishing potential of expansion for knowledge and the warranty of validity of its development, but at the same time in doing so they translate their theory of proof in a presupposition that they are simply not geared neither to express without circularity nor, by consequence, to justify, since it is that which make them possible in every moment, as Frege noticed regarding the laws of inference in his *Begriffsschrift*.

Thus, according to Cassirer's criticism to Hilbert's Finitism and axiomatic method, that which operations of computing devices are is (1) undefinable and (2) unjustifiable. Allow me to illustrate why this is the case for computational devices too, since it is not the case that a trivial inference can be done from "they use formal languages" to "they have the flaws Cassirer indicates about Hilbertian finitist procedures".

(1) If it is impossible to define, for example, that information can describe any process of communication in terms of probability of the transition to a certain event from a certain series of events, or as it is the case of Hilbert's numerals that signals are numbers, then information theory as well as numerals are a bunch of writings that has no point in being called knowledge. For a relation is all that it is taught in the cases of both informatics and finitist mathematics, but if this is a relation between signs only, it becomes a formalism, a purely arbitrary game between signs. As Frege puts it, «the fact that we concern ourselves at all about the reference of a part of a sentence indicates that we generally recognize and expect a reference. The thought loses value for us as soon as we recognize that the reference of one of its part is missing […] because, and to the extent that, we are concerned with its

truth value»[383]. For something to be called rational and knowledge, the desideratum is that it makes something more intelligible in some sense (more defined, more clear, necessary given some conditions, manifest in its genesis and many more), but unless we are interested in the *significance* of a production – in the sense we are interested in the value of tragic plays, for example – all of these are properties that make sense only regarding statements about referents of acknowledged epistemic interest, such as numbers, reality, moral and many others. Thus, the first problem I want to show is that these systems cannot define on their own the interpretation that makes them interesting, whereas this is a central point, especially in knowledge. The previous paragraphs provide sufficient material for documenting the fact that the behavioral procedures of computing machineries do not make available any reference for they to be self-interpreted: they are merely executed.

Let the reader recall the explanation of how TMs and computers function in §1 and §4: they have a purely behavioral approach to instructions; specifically in classical computing devices this means that a certain output must correspond to certain input given a certain antecedent machine state. This means that, as in formal systems, instructions can be considered only as inputs, and as such they cannot be just stated, they must be executed in every moment they are received, unless they are just stored. I have shown that as soon as they leave the storing location, they are transmitted to the executive components, and this is the whole story. We can therefore conclude that these instructions either are executed, but this means that their understanding – which is not there in the sense this word is commonly used – must be presupposed, and this presupposition consists of structural and operational features that the machine cannot consider in a theoretical fashion; or they are saved as a physical status, whose interpretation consists of becoming a certain input in a defined procedure, therefore in this case too there is no possibility of taking into account the fundamental laws or the referents of the instructions per se, since this would amount to the machine be aware of its own instruction set architecture and memory content (states), which is false: machines act accordingly, but *have no descriptive capacity of their own states*, through which they should posit their own structure and its consequences as the referent of their own operations. If I am right, then computing devices, since they use information-processing and the alleged semiotic sense of being interpretable, can neither define the referents of their own operations, nor produce their laws, therefore they are not independent, at least regarding the perspective of knowledge, since they are unable to define that which make their eventual symbol-processing significant and justify it.

[383] Frege 1892/1960, p.63.

What difference does this make for the problem of representation and computational explanation of mental activities? It makes more than one. First the conclusion implies that, either if considered as information-processing or if considered as formal manipulations, computational processes fail in principle in implementing a relevant feature of mental activity, namely the possibility of (A) defining the reason for being interesting regarding the activity itself, (B) defining the principles of their own operations in a non-tautological way. Of course, it can be objected that humans too cannot do both. This objection is manifestly false, for I just did define the reason for which (hopefully) the symbolic production the reader is observing may be of interest, since I can count on the *symbolic competence* that all human beings share, i.e. the capacity of seeing these signs once as material items, once as sense-vehicle/reference-vehicle. Regarding point B, humans have further strategies available, which computing machines have not: for, example, as Aristotle did for the non-contradiction principle and the law of the middle excluded it is possible to show that they cannot be denied without performing a self-contradiction or giving up the whole possibility of speaking; as Kant did for causality, it is possible to demonstrate that fundamental conditions of intelligibility are indeed conditions of intelligibility.

Second, regarding specifically the problem of representation, the conclusion shows that even if one is willing to buy the semiotic account of meaning and the informatic idea of communication (meaningful transmission), nevertheless Cassirer's and Frege's criticisms show that the resulting activity is unable to be self-contained: it requires an external agent for it to be interpreted, therefore reference is something that lies outside the reach of computing machines, and whatever one thinks about that which symbols and representations are, I hope the necessity of reference will not be put in question, since otherwise I cannot see how one can differentiate material configurations from signs and representations.

Third, finally, the survey on the debate about the foundation of mathematics shed some light on the problem of reference, which is close to Harnad's symbol grounding problem[384]. The rules allowed in formal systems are rules of substitution, the operational approach of computing machines consists of replacing signals over series of signals, and this correspond to the associative way of dealing with functions I mentioned in the previous paragraphs. Such an associative way implies that every attempt of positing a content as sending to another content is frustrated, because the former must be replaced by the latter

[384] See Harnad 1990. I write that the following statement is "close" and not equal because Harnad in his paper gives the fact, but not the reason: he says that Searle's problem implies a more serious one, but he does not explain in detail the roots that make Searle's criticism relevant.

in a conservative and asymmetrical way for a representation to be formed (see *Chapter Fourth*, §1), but this is exactly the *contrary* of how computational machines operate. In the end, even if multiple contents are stored, just some kind of behavioral relation can be posited between the two, which is a causal/procedural connection at the very best, but this is not reference in the proper sense involved in representation, since not every representation implies a causal connection, nor referring is a practice, but a *property* relative to some object only when it is in a proper relation with an adequate subject (see again *Chapter Fourth* for more on this).

(2) Inability to define the referents of its own production, and impossibility of expressing in a non-circular and non-behavioral way the fundamental rules of its own operating, causes in turn computing devices to be dependent on external agents for their rationale to be accounted. In fact, they not only cannot even consider their own operating, but just act accordingly; they also must presuppose their (of relations and relevant rules) being stated in a proper form in order to be able to operate over them and eventually to define them in a descriptive way (which I have shown does not happen in any case, since operations are executed and never considered). If this is the case, they cannot account for themselves, since the best they can do consists of being operationally consistent with themselves, but consistency grants neither meaningful reference nor validity of inferences, unless one presupposes that the beginning assumptions are already specular to the problem they must lead to solve and that the basic rules are themselves consistent – as Hilbert does. Thus, these systems cannot account for the validity of their procedures, just follow them.

To sum up, even under the conditions for ascribing signs and meaning in a certain technical sense to signals and information-processing operations, computing devices with their programs and informatic signs suffer the objections that Cassirer stated to Hilbert's Finitism and Constructivism in philosophy of mathematics and geometry: they are unable of talking about symbols, programs and referents within their computation, they do not conceive, they just act accordingly, hence their being computational, even if actual, is an *extrinsic presupposition* and something beyond their reach, so that they can neither justify themselves, nor provide their own interpretation. Thus, any claim of rationality or representational capacity for computing devices falls into the twofold contradiction of, on the one hand, being unable to explain why these operations, products and so on are relevant for the theoretical – amongst other – interest they would like to explain and represent; on the other hand, of these computational subjects being unable to define these very same referents they are explaining or manipulating, so that neither symbolic competence,

symbolic possession and by consequence representational activity can be ascribed. Computational subjects therefore would lack all the conditions of possibility for an intrinsic productivity of representations, and even signs. I think that, if true, the objection at issue would be a serious impairment for positing computational subjects as explanatory model of human mental activity.

Chapter Third. Connectionist Computation

In this chapter, first I discuss the general rationale behind connectionist systems by an exam of that which is considered the conventional beginning of the connectionist model of computation, i.e. McCulloch and Pitt's theoretical neural network. Then I provide some references for having an outline of the contemporary strategies for implementing connectionist computation. After this has been done, I discuss which sense artificial neural networks compute, if they do. In the conclusions of the chapter, I show that these devices have the same features that make relevant the previous criticisms, namely: they are information-processing devices; they compute, i.e. their operations and components can be accounted only in terms of executing an effective procedure over an internal flow of signals, and that they are behavioral exactly as their classical counterpart. A novelty in my position is that, according to my analysis, at least some of these machineries manipulate "symbols" in the same sense as their classical counterparts do, since in both cases "being a symbol" is just a functional role that can be expressed in informatic-semiotic terms. Even if it is a novelty relatively to philosophical debate, it is consistent with the common treatment of connectionist devices in scientific handbooks.

§1. Historical Perspective: McCulloch and Pitts' Work and Contemporary Artificial Neural Networks

1.1.Exposition of the Structure of MP Nets

Piccinini expresses accurately the essence of the model they proposed: «[…] McCulloch-Pitts' computational theory rested on two principal moves, both of which are problematic. On the one hand, they simplified and idealized neurons, so that propositional inferences could be mapped onto neural events, and vice versa. On the other hand, they assumed that neural pulses correspond to atomic mental events endowed with propositional content»[385]. Piccinini refers specifically to the starting set of assumption at the beginning of the paper in question, which are neurophysiological assumptions[386]:

- "all-or-none" character of neural activity: this property is defined as follows in Kandel et al. 2013, p.33: «the action potential[387] is all-or-none: stimuli below the threshold do not produce a signal, but stimuli above the threshold all produce the

[385] Piccinini 2004, p.206. Italics in the text.
[386] McCulloch e Pitts 1943, p.118.
[387] Cfr. Kandel et al. (2013), p.23: «action potentials are the signals by which the brain receives, analyzes and conveys information». This definition applies information theory, which is the standard position in treating the topic of semantic content in brain. Anyway, the definition individuates that electrochemical charge that surpass the threshold value for activation and that lies at the basis of cerebral activities.

signals of the same amplitude». It means that either the sum of afferent stimuli to a neuron reaches a value beyond the threshold potential, and in this case always units of signals of the same kind and homogeneous are produced, or nothing happen, despite the signals received;

- for the threshold potential to be surpassed, which is constant, contemporary afferent signals must be discharged towards a certain target neuron[388];
- the only significant delay comes from the transition from one synapse to another;
- inhibition signals have an absolute effect: just one is sufficient to prevent any activity from the targeted neuron[389]; Rojas shows that the authors are right in this claim: absolute and relative inhibition prove to be equivalent[390];
- «the structure of the net does not change with time». Authors assumed invariance of neural net since «for nets undergoing both alterations[391], we can substitute equivalent fictitious nets composed of neurons whose connections and thresholds are unaltered»[392]. Rojas shows that the authors are right in this claim: weighted and unweighted nets are equivalent[393]. Despite this there are differences: if a network is unweighted, in order to adapt its output, the disposition of the unities (topology of the net) will need to be altered, whereas weighted network can use algorithms for adjusting the weights[394].

These assumptions are idealizations, since the last two are false from the standpoint of biology, but the authors are aware of this: they stipulate these assumptions because they judge them irrelevant regarding their conclusions. The second assumes as fixed that which may vary: threshold potential is higher during the refractory period of the neuron, during which it cannot produce any new action potential, or it can be lowered or augmented under some circumstances.

Piccinini deduces the finiteness of the states in the net from the third assumption, which is

[388] Cfr. McCulloch e Pitts 1943, pp.115-116.
[389] Cfr. Minsky 1967, p.34: «our "inhibition" is here absolute, in that a single inhibitory signal can block response to a cell to any amount of excitation».
[390] Rojas 1996, pp.39-40.
[391] They refer to permanent changes either of threshold values or neural connections.
[392] McCulloch e Pitts 1943, p.117.
[393] Rojas 1996, pp.38-49.
[394] Rojas 1996, p.46: «the first clear separation line runs between weighted and unweighted networks. It has already been shown that both classes of models are equivalent. The main difference is the kind of learning algorithm that can be used. In unweighted networks only the thresholds and the connectivity can be adapted. In weighted networks the topology is not usually modified during learning (although we will see some algorithms capable of doing this) and only an optimal combination of weights is sought».

relevant for equating McCulloch and Pitts nets to Turing Machines[395]. Minsky states that the feature of finiteness derives from the first assumption and the assumption that the discharging time is synchronized between all units, namely the assumption that the succession of time $t_1, \ldots, t_n$ is counted in the same way by all the units, so that it is possible to posit that only a certain set of neurons composing a certain layer fire at time t[396]. That which is supposed to make McCulloch-Pitts nets comparable to TMs is not precisely stated in the original paper (I will discuss this later), but when the author define the notation for their nets, actually there are the factors indicated by Minsky, therefore I will follow his reading and I will not acknowledge a specific importance to the third assumption, even if Piccinini's reading is consistent, since the delay between different neurons can contribute to make each transmission discrete, if the delay is assumed as a scan of time and the state of the net are considered accordingly.

I will follow Piccinini regarding the simplification of the formal notation applied in the original paper, which commenters agree in defining unnecessarily complex and in individuating some errors in its application both logical and typographical[397], so that it is preferable to refer to some accredited reconstruction. I will adopt Minsky's style for representing *logic units*.

Each unit is named "c_i", where the index designates the number of the unit. The notation "$N_i(t)$" indicates that the unit fires at time *t*. In the original paper, units that fire at the same time are represented on the same vertical line. Each unit fires only if a certain threshold potential is reached, indicated in fig.5 by letter *x* on the left side of the unit, which is the side with afferent "synapses", whereas the other is the side with the "axon". If $x = 0$, then the unit is always active; if the signal is excitatory, namely it contributes to firing of c_i, then there is an arrow at the end of the connection, in case it is inhibitory, there is an empty point instead. Propositions are defined as the set of the behavior of all units that fire at time $t - 1$ and are afferent to unit $N_i(t)$. Thus, propositions are posited as if they *belong* to unit $N_i(t)$, since it is its firing that "pronounce" the proposition which the other units make possible. Neurons in visual cortex apply a similar strategy: there are neurons which fire only if certain conditions are

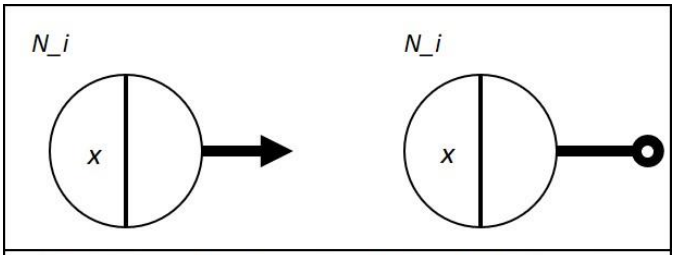

Fig.5. Schema of exciting neuron (left) and inhibitory neuron (right), according to notation in Minsky 1967.

[395] Piccinini 2004, p.192.
[396] Minsky 1967, pp.33-34.
[397] See Fitch 1944; Minsky 1967, p.34; Piccinini 2004; Schlatter e Aizawa 2008.

met; the principle is the same. Well-formed formulas in this calculus are composed of[398]:

1. name of units;
2. Boolean operators, plus the function "*S* (P) (t)", which asserts that a predicate "P" which holds for a number holds for its predecessor as well;
3. standard quantifiers.

Let then be the notation for an expression stated by c_i to be the set of "assertions" made by all units up to c_i from an arbitrarily distant (in enumeration, not in space) unit c_g. In this case, $N_i(t) = Expression_i = [N_{i-g}(t-1), \dots, N_{i-1}(t-1)]$. In this example, operators are omitted, and it is assumed for the sake of exposition that all the units before $N_i(t)$ fire at the same time, but this is not necessary. In figure 6, I introduce through Minsky 1967 some examples of how these units can be combined in order to realize three common operators and how their formal expression can be written in the notation outlined. If you imagine hardwiring the properties described by these units through suitable circuits, then you have a logic gate. Minsky 1967, pp.46-51 also shows how to construct decoders and encoders through McCulloch-Pitts nets.

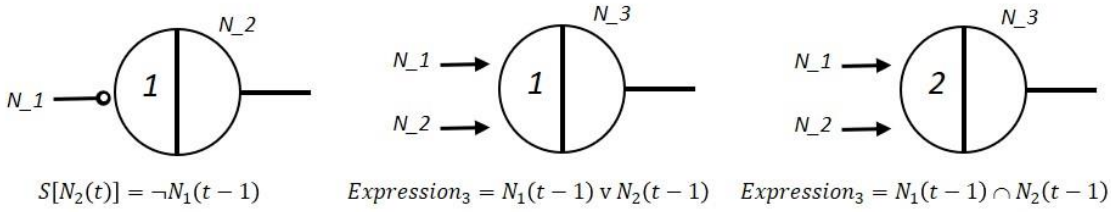

Fig.6. Example of three logical operators as stated by McCulloch-Pitts units, with their expressions in the stipulated notation. See Minsky 1967, p.33 for further examples.

Once they defined model and formalism, the authors proposed two problems: «our central problem may now be stated exactly: first, to find an effective method of obtaining a set of computable **S** constituting a solution to any given net; and second, to characterize the class of realizable **S** in an effective fashion. Materially stated, the problems are to calculate the behavior of any net, and to find a net which will behave in a specified way, when such a net exists»[399]. Here I am not interested in the solution to these problems, but in moving the first steps toward a definition of the sense in which those nets compute.

This has been a debated topic since the publication of the paper. For the authors state that these nets "obviously compute", but actually they did not provide any formal or informal

[398] McCulloch and Pitts 1943, pp.118-119.
[399] McCulloch and Pitts 1943, p.119.

evidence that this is the case[400]. Precisely, the authors write that «it is easily shown»[401]:

1. that if these nets are equipped with tape, scanners connected to afferent units and suitable efferent connections from other units for sensorimotor activities, then those nets can compute *only* Turing-computable sequences;
2. that if these nets are equipped with tape, scanners connected to afferent units and suitable efferent connections from other units for sensorimotor activities, then those nets are *Turing-complete*, namely they can compute *all* Turing-computable sequences;
3. that *nets with circles can be computed by nets* equipped with tape, scanners connected to afferent units and suitable efferent connections from other units for sensorimotor activities;
4. that nets can compute *some* Turing-computable numbers, even if not equipped with tape, scanners connected to afferent units and suitable efferent connections from other units for sensorimotor activities.

Moreover, they add the claim that «this is of interest as affording a psychological justification of the Turing definition of computability and its equivalents, Church's λ-definability and Kleene's primitive recursion: if any number can be computed by an organism, it is computable by these definitions, and conversely»[402]. Otherwise stated, McCulloch and Pitts are claiming they justified, by providing a plausible idealization of biological neural nets, the fact that the Church-Turing thesis has a biological ground: every computable function is Turing-computable because brain is a Turing-complete device, and since Church's and Kleene's formalism for computing are Turing-computable, the authors claim that they can be computed by connectionist organisms[403].

Any of the previous claim is easy to be shown, but despite this, the paper can be regarded as the first attempt of connecting computation and mind in a consistent and detailed manner[404].In *Representations of Events in Neural Nets and Finite Automata* (1951), Stephen Kleene will be able to demonstrate exactly that which these nets actually compute. Kleene defined two key concepts:

[400] Piccinini 2004, p.198; Schlatter and Aizawa 2008, p.247.
[401] McCulloch and Pitts 1943, p.129.
[402] McCulloch and Pitts 1943, p.129.
[403] See also Piccinini 2004, p.199.
[404] For example, Cowan 1990, p.76 has enthusiastic words for the historical importance of McCulloch and Pitts' work. See also Piccinini 2004, p.197: «by stating that McCulloch-Pitts net computed, this passage provided the first known published link between the mathematical theory of computation and brain theory. It was a pivotal statement in the history of computationalism».

- *finite-state automata*: they are sufficiently individuated by the fact that once that their initial state, their input and the description of the machine have been provided, their behavior can be exhaustively described[405].
- *regular languages*: the informal idea is simple: basically, every language whose well-formed formulas shows recurrent patterns in consequence of the finite set of rules that generate all the expressions is a regular language.

Following Minsky's explanation[406], I present the formal definition of regular expressions. First a base is posited, let say letter *x*. The letter symbol alone is a wff and a sequence in the language *L*. The *repetition* of *x* for *n* times (for each $n \geq 0$), indicated as "*x**" is a wff too and another allowed sequence in the language *L*. Sequences formed by *concatenation* (juxtaposition) and by *union*, that is by combining all possible sequences in a language with all the possible sequences in another language, are *regular expressions*. Thus, for example, we have regular sequences as: "a", "$aaaaa$" and similar, if there is just one symbol in the language *L*; "$aaab$", "abababababaaaa", and so on, if there is more than one. We have instead regular expressions as "$xxx \vee xx$ " or "$(a \vee \mathrm{ab}) * \mathrm{a}$" or "$(ab) * b$".

Regular expressions can be defined more formally as follows:

1. first, consider *the following definitions*: given symbols "$s_1, \ldots, s_n$", and assuming "ε" as the symbol of the empty sequence, we have "*L(E)*" as denoting the language composed by the set *E* of all regular expressions, formed as follows;
2. *by concatenation*: for two expressions E and F, $L(EF) = \{ef : e \in L(E), f \in L(F)\}$;
3. *by union*: for two expressions E and F, $L(E \vee F) = L(E) \cup L(F)$;
4. *by repetition*: for an expression E, "$L(E *) = \{\varepsilon\} \cup \{L(E)\} \cup \{L(EE)\} \cup \ldots$".

Basically, that which has been done is:

- defining a base, namely a set of primitive elements;
- defining rules of combination by repetition, concatenation and union;
- positing a restriction axiom: all and only that which is composed by the base and the application of the rule of repetition to it is a member of the language.

Kleene in his theorem 6 demonstrated that all events taking place in a finite automaton as

[405] Kleene 1951, pp.75-77; Minsky 1967, pp.11-12; Rojas 1996, p.43.
[406] See Minsky 1967, pp.71-74.

McCulloch and Pitts' nets can be described by regular expressions (regular events)[407]; that all regular events are primitive recursive in theorem 8[408]; that not all events which are primitive recursive are regular events in the Appendix 3[409]. Thus, Kleene showed that McCulloch and Pitts' nets can compute a subset of primitive recursive expressions, called regular languages, therefore those nets can compute *a specific class of Turing-computable functions*.

1.2. Realism or Instrumentalism Regarding Formal Languages in MP Nets

According to the first three paragraphs of McCulloch and Pitts' paper, computation of those nets seem to be a matter of *describing* them as computing something, since the formalism introduced by the authors describes the activity in the net, but they do not seem to understand these languages as the proper object of the nets, since they talk about representing or conceiving the nervous activity in terms of propositional calculus. This up to the line where they advance the thesis that these nets can be used to implement the control unit of Turing Machines, and up to the paragraph named "Consequences", in which they discuss the explanatory value of their theory. There the focus change: formal computation is no more a descriptive tool, but a realist, theoretical explanatory tool, something on the same stance of Wiener's application of information and mathematical theory of feedback to the problem of control in living beings. This change in attitude is relevant for understanding how something as a representation can be ascribed to this kind of nets in their original context, in which sense they compute, and it is also important regarding the computational answer to Behaviorism and its three questions, as formulated in §2 of *Chapter First*. Of course, there have been many changes to the theory of neural nets, hence this is just an historical perspective, which will be updated in the following paragraph.

The strength of McCulloch and Pitts' work is that, for the first time in history, someone was able to formulate a propositional calculus compatible with the brain and suitable to be computed in Turing's sense. This makes reductive materialism regarding the mind-brain relation an experimental hypothesis over than a metaphysical thesis. In the original paper, the arguments of each proposition are the activations of each units together with their structure, whereas the composed propositions are the series of activations in the net, so that there is a one-to-one correspondence between units and parts of the propositions. In this way, mental content and physical structure are described by one theory, mathematically

[407] Kleene 1951, p.80.
[408] Kleene 1951, p.90.
[409] Kleene 1951, p.95.

structured and at the same time open to be integrated by psychology or neuroscience. This direct identity carries its own difficulties, but it exerted a great charm to those working in the raising field of automata theory, especially after the integration of logic gates in the design of computers by Von Neumann.

An historical approach to the paper supports the realist reading in the paragraph "Consequences". McCulloch was theorizing and seeking from the '20s a fundamental unit of the psychic event he called "psychon"[410]. The formulation of this theory shares two features of the propositional calculus introduced in the paper at issue:

- each "psychon" is a proposition related to its causal antecedent and collocated in an antecedent time. This corresponds to setting that the activation of c_i expresses in *t* the state of units from c_{i-1} to c_g in *t-1*;
- the content of each "psychon" is transmitted to the next. This corresponds to the facts that it is the connection between the units to structure asserted by the final unit and that the assertion is composed by the sum of the contents asserted by each unit, as it is showed in fig.6.

The comparison is relevant because at the beginning of *A Logical Calculus…* the authors declare that «many years ago one of us, by considerations impertinent to this argument, was led to conceive of the response of any neuron as factually equivalent to a proposition which proposed its adequate stimulus. He therefore attempted to record the behavior of complicated nets in the notation of the symbolic logic of propositions»[411]. McCulloch had given a physiological interpretation of the intuition described above since about 1929[412]. The genesis of the theory thus suggests that the identification between mental content and physical processes must be understood in a strong, realist sense. Despite this, the first pages of the paper seem to witness against this interpretation[413]. Minsky also understands their work according to an instrumental line of thought: «the representation of the analysis of the logic of the situations that arise in any discrete process, be it in brain, computer or anywhere else»[414].

Kleene's interpretation may help in giving a consistent account. Even if it is neutral regarding this specific problem, Kleene's description is closer to a realist interpretation,

[410] Cowan 1990, pp.75-76; Piccinini 2004, pp.177-178.
[411] McCulloch and Pitts 1943, p.117.
[412] Piccinini 2004, p.179.
[413] See for example McCulloch and Pitts 1943, p.117: «the "all-or-none" law of nervous activity is sufficient to ensure that the activity of any neurons may be represented as a proposition».
[414] McCulloch and Pitts 1943, p.117.

since the components of the propositions – the inputs asserted by each unit – are explicitly identified as *events* in the neural net[415]. Thus, each argument of the operators that form propositions is something that concretely *takes places into the net*. If this is the case, it is possible to posit a situation similar to that of the classical models of computation: even if the events in the net are just describable as computing a formal language, the necessity of accounting their activity and their rationale may be sufficient for ascribing computational properties and, in this case, this is also sufficient for acknowledging the kind of explanations that the paragraph "Consequences" proposes.

The model of explanation at issue goes as follows. There is a direct relation of identity: all that brain produces depends on the activity within it, and this activity is now determined as computational. That this identity is not understood as instrumental, this time is clearly stated in at least three passages:

- «the role of brains in determining the epistemic relations of our theories to our observations and of these to the facts is all too clear, for it is apparent that every idea and every sensation is realized by activity within that net, and by no such activity are the actual afferents fully determined»[416]. There, the authors propose a *structuralist* conception of knowledge: not only the activity, but also the distribution of the afferent neurons determine the content of internal data. Relevant lesions in the net, insufficient specification or underdefined processing account for cognitive disorders such as paraesthesias and hallucinations;
- «to psychology, however defined, specification of the net would contribute all that could be achieved in the field – even if the analysis is pushed to ultimate psychic units or "psychon", for a psychon could be no less than the activity of a single neuron»[417]. Thus, the authors call also for a reduction of psychology to a form of computational, theoretical neuroscience. It does not matter anymore what field psychology vindicates for itself as a science of the behavior of the living being, all the answers will be provided by the individuation of the relevant features of the net;
- the authors also claim that «thus both the formal and the final aspects of that activity which we are wont to call *mental* are rigorously deductible from present neurophysiology»[418].

[415] Kleene 1951, p.9
[416] McCulloch and Pitts 1943, p.131.
[417] McCulloch and Pitts 1943, p.131.
[418] McCulloch and Pitts 1943, p.132. Italics in the text.

The picture that results from these claims is the idea that mental events are "*obiecta*" and "*facta*". I mean, respectively, that according to this theory mental events are (I) something that is given into material existence, mind is out there as all other objects, and as such it can be observed: mind is completely equivalent to its material existence. This may also sound trivial, but the implications are not: as McCulloch and Pitts remark, psychology is ruled out, there is no need of an abstract and descriptive account, if the efficient causation of the process, content included, can be observed. Of course, as Piccinini notices, «McCulloch and Pitts' project was not to systematize and explain observations about the nervous system – it was to explain knowledge and other mental phenomena in terms of mechanisms that resembled neural ones»[419], i.e. their explanatory project concerns living operations, not anatomical, purely structural account: it is a mix of the two, but always addressed to the dynamic and functional aspects of mental activity. Nevertheless, their account is relentlessly reductionist. Mental events are also "*facta*" (II) in the double sense of being something that knowledge find as already there, complete and self-sufficient, and of being a concrete event. This also may sound trivial, but it is not. For according to this thesis, that which is ruled out this time is the ordinary, discursive content of knowledge and it is denied that the subject has any to do with his/her own thought. As Nietzsche stated once in a metaphor, there are some thinkers that are the soil for their thoughts: they just receive their thoughts once they are grown out from the ground, but they play no role in making them, thoughts follow their own internal economy, and we just grab them once they are ready[420].

To sum up, McCulloch and Pitts understand their theory with a strong realist stance, a realism whose modality of explanation consists of positing a completely reductive identity between *flow of activations in the net and the processing that the structure produces over these activations, together with the resulting mental contents and features*. In this theory, mind disappears from the picture as something scientifically irrelevant, but it is not annihilated, it is just transferred into the material processing; in the words of the authors: «since that activity (my note: of the units in the net) is inherently propositional, all psychic events have an intentional, or "semantic" character»[421]. This is the reason for the heading, "a calculus of ideas *immanent* in the nervous system".

The reader may appreciate that this stance grants an answer to the questions raised by Behaviorism on mind: observability, being a suitable object of experimental practice, and

[419] Piccinini 2004, p.203.
[420] It is in *Daybreak*, §382.
[421] McCulloch e Pitts 1943, p.131.

being causally relevant of mind. Through the transition from a mental consciousness to a mental event, mind becomes observable, constructible and therefore suitable to be object of experiments that check the consequences of the construction; mental causality is now structural and exerted as the effect of procedural rules and interaction of physical events, hence something scientifically respectable and describable.

What kind of explanation does this model bring? The one Von Neumann efficaciously described during his talk at the Hixon Symposium: «it is perfectly possible that the simplest and only practical way actually to say what constitutes a visual analysis consists in giving a description of the connections of the visual brain. […]. We have no right to assume that the logical notations and procedures used in the past are suited to this part of the subject. *It is not all certain that in this domain a real object might not constitute the simplest description of itself*. […], It is, therefore, not at all unlikely that it is futile to look for a precise logical concept, that is, for a precise verbal description of "visual analogy". It is possible that the connection pattern of visual brain is the simplest logical expression or definition of this principle»[422]. Von Neumann's position in this passage can be summarized as follows. There are objects, as automata or human brain or even parts of the latter, that are too much complicated to be discursively described, both in their functioning and their structure. It is therefore possible that the best, rigorous theory about them consists of a precise conception of that which they are and of the course of their activity as it is, once provided that it is described in a suitable scientific form. As I said, the presupposition of this thesis is that conceptual contents can be observed in their existence insofar as they must be or must be grounded on objects: the thought must offer itself in the same way as any other object does. This is the only sense in which it is possible to account passages in McCulloch and Pitts 1943 as «with determination of the net, the unknowable object of knowledge, the "thing in itself", ceases to be unknowable»[423] and «mind no longer goes "more ghostly than a ghost"»[424] (a paraphrasis of a famous quote by Sherrington).

Computation can be ascribed in the same way as it has been ascribed to classical computing devices in §5 of this chapter. Since the MP nets are designed for computing, since their operating, properties and effects cannot be understood unless reference is done to computational activities and properties, it is necessary to ascribe computation to these nets too. They are not only describable through the formalism that are sometimes applied to other computing devices, they also have features that depends on mathematical and logical

[422] Von Neumann 1951, p.24. Italics added.
[423] McCulloch and Pitts 1943, p.131.
[424] McCulloch and Pitts 1943, p.132.

properties. I will show the correctness of this claim in the next part of the paragraph.

Concerning the ability of representing, as for any other activity, representing is now a matter of being able to operate over inner states that constitute the content of the understanding of the outer reality, as in the case of classical computing devices, with the difference that *representation has a distributed existence over the net*: namely, its *logical structure* lies in the order and quality (the class they belong to) of connections, its *content* is provided by the flow of signals that is ongoing in the net, as in the case of informatic signals in classical devices. Connectionist computation as it is conceived in MP nets has become outdated under significant respects – especially the organization of the model and the mathematical description – but the idea of distributed existence and the explanatory model they presuppose is still at work. Before venturing on further analysis on the problem of representation in connectionist systems, it is preferable to discuss contemporary models of connectionist computation, the sense in which they compute, and how they work.

1.3. An Outline of General Principles

Scientific research on connectionist computation has made enormous progresses, and other are still ongoing. Therefore, it is hard to produce an exhaustive classification, even if some core points can be fixed. I will move from them to built progressively the relevant details for the present purposes, namely understanding in which sense connectionist devices compute and in which sense, if any, they represent. As I anticipated, I am going to defend the thesis that the results in §§5-7 of the previous chapter hold for connectionist computing too. This may sound somehow counterintuitive, since usually philosophical literature draws a sharp distinction between serial and connectionist computing. This is correct, since there are crucial differences regarding architecture, functioning and properties, but there are also points of contact between the two, both theoretical and concrete, and I am going to show that they justify the extension of the results in the previous chapter to connectionist systems.

Artificial neural nets (ANN from now on) are produced both through *computer simulation* and *specialized hardware*[425]. Notice that the use of specialized hardware has never been used in recent times for very large structures: simulated solutions are widely preferred, since they offer several practical advantages, even if at the cost of some extra

[425] Some chips have been developed for hardware implementation of neural nets, as for example memristors, spintronic memories, threshold switches. For example, hardware ANN are described in Jain, Mao and Mohiuddin 1996; Zhu, Milne and Gunther 1999; Yang, Strukov and Stewart 2012; Grollier, Querlioz, Stiles 2016; Jeong and Shi 2018. Software implementation in simulation environment are reviewed for example in Skrzypek 1993. Further software models as The GUI, GENESIS or The Nest are reviewed in Brette, Rudolph et al. 2007.

computational resource. However, the two solutions are not mutually exclusive: some of the components in nowadays computers are compatible with the purpose of simulating ANNs, even if they have some limitations both in the functions they can accomplish and, above all, of computational power[426]. Despite this, today clouding technologies and increased computational resources of some devices fulfill the needs for a simulated implementation.

Artificial neural nets are constituted by *units* or *artificial neurons*, which forms the *nodes* of the net. Those units *are described by functions* (or, in the case of simulated networks, they *are* functions, still standing that which has been said about this in *Chapter Second*). Rojas 1996 explains that: «if we are computing with a conventional von Neumann processor, a minimal set of machine instructions is needed in order to implement all computable functions. In the case of artificial neural networks, the primitive functions are located in the nodes of the network and the composition rules are contained implicitly in the interconnection pattern of the nodes, in the synchrony or asynchrony of the transmission of information, and in the presence or absence of cycles»[427]. This confirms also that a general idea in McCulloch and Pitts' work has been preserved: the composition of the proposition and the steps of the operations are embodied in the structure of the net.

It is better to stipulate Haykin's and Rojas' notations for relevant parts of the units, so that exposition will be briefer and more precise. Thus, let:

1. "$x_1, \dots, x_m$" designate the *input signals from connecting links*;
2. "$w_1, \dots, w_m$" designate the *corresponding weighting value*;
3. if it is written "x_{kj}" or "w_{kj}", it means: the input/weight for the input *j* in unit *k*, which is the immediate successor;
4. "$u_1, \dots, u_m$" designate the *output of the integration function "g"* (or Σ in Haykin) for the unit *m*;
5. "$y_1, \dots, y_m$" designate the *output of the activation function "f"* (or φ in Haykin) for the unit *m*;
6. "b_k" designate the *bias factor over the output* u_k for the unit *k*.

The units are composed as follows[428]:

- A set of *connecting links*¸ equivalent to biological synapses, which have a "*weight*"

[426] Cfr. Dias, Antunes and Mota 2004.
[427] Rojas 1996, p.23. See also Haykin 2009, p.10.
[428] Rojas 1996, p.23; Haykin 2009, pp.10-11.

or "*strength*" on its own, which is indicated by "*w*". The input cells, which are those directly connected with the data over which ANN performs its computation, have connecting links called "*synaptic links*", and their behavior results from a linear input-output relation, namely their signal is multiplied for the weighting factor (see next point) and then directly transmitted[429]. Hidden units (units that perform intermediate operations) or output units have instead connecting links called "*activation links*", whose behavior results by the application of an activation function, which is discussed below[430]. It is presupposed that each connecting links transmit always in the same direction, and this property is called "*unlimited fan-in property*"[431]. The "*unlimited fan-out property*" is instead the principle according to which content is transmitted to every successive step without being dependent on any function of transmission stated through the efferent connections[432];

- an *adder*[433] or *integration function*[434], namely an input part of the unit which consists of computing a function that sum all the inputs according to their weighting factors. Generally, a linear combination function is applied, that is a summation;
- the *bias factor*[435], which is a parameter that modulates the value *u* resulting from the activation function, by increasing it if *u* is positive, whereas it lowers the value if *u* is negative. Bias factors are set at random at the beginning. The bias works exactly as a prejudice: if the outcome of the computation has a conformity given certain conditions, the bias value is strengthened so that the firing of that unit gains an advantage over other outputs, and the opposite happens in the opposite case. The bias is equivalent to *the threshold value of the unit*: if it is negative, it requires *u* to be higher; whereas if it is high, even if *u* is lower the unit is more likely to fire;
- an *activation function*, i.e. some computation over the output of the integration function. Haykin explains that the result of this function consists of recollecting the values produced by the function *g* into a normalized value , that is a value included in some allowed interval. Usually, the range is set as {0;1} or {-1;0;1} or {-1;1}. There are many kinds of activation functions. Haykin provides the following list:

[429] Haykin 2009, p.15.
[430] Haykin 2009, p.15.
[431] Rojas 1996, p.31.
[432] Haykin 2009, p.16.
[433] Haykin 2009, p.10.
[434] Rojas 1996, p.31.
[435] Rojas 1996, p.61; Haykin 2009, p.11.

- *threshold function*[436]: it is simply the mathematical translation of the "all-or-none" character of neural activity. Units that apply this function are in fact also called "McCulloch and Pitts Neurons". It expresses rules of the following form: $\varphi(u) = \begin{cases} 1\ if\ u \geq 0 \\ (\ 0\ if\ u < 0 \end{cases}$;

- *sigmoid function*: it is so-called because of its typical shape. It expresses rules of the following form $\varphi(u) = \begin{cases} 1\ if\ u > 0 \\ 0\ if\ u = 0 \\ -1\ if\ u < 0 \end{cases}$:;

- *stochastic model of a neuron*: it allows probabilistic calculation within *f*. It expresses rules of the following form: $\varphi(u) = \begin{cases} +1\ with\ probability\ P(u) \\ -1\ with\ probability\ 1 - P(u) \end{cases}$.

Differences in the net largely depend on the activation function they are implementing, which Rojas calls *primitive functions*[437]. The reader can understand from this exposition that basically *ANNs repeatedly perform the same function over and over across their layers*. Anyway, ANNs do not this as a mere repetition, instead there are several possible dispositions of the layers and connections between units, over than many ways of defining the families of functions summarized, that allow the emerging of relevant properties and affect which is the input of the next function.

Something general may be introduced about the question of how the units are connected and how the layers may be arranged, so that the subsequent analyses will result more familiar. Concerning connections, the main difference is between *feedforward* connections and the already mentioned *feedback* connections. In the context of ANNs, a "feedforward" system is a structure as McCulloch and Pitts' nets without cycles: each layer sends its output to the successive one, without any loop[438]. The "feedback" systems are precisely described by Haykin: «feedback is said to exist in a dynamic system whenever the output of an element in the system influences in part the input applied to that particular element, thereby giving rise to one or more closed paths for the transmission of signals around the system»[439]. Otherwise stated, a net whose layers may transmit inputs to successive layers, previous layers or the same units that send the output, in such a way that the result of a

[436] Haykin 2009, pp.13-14. See also Rojas 1996, pp.60-61. Rojas calls Haykin's form of exposition the *geometrical interpretation* of threshold functions.
[437] Rojas 1996, p.23.
[438] Cfr. Rojas 1996, p.29-31. Haykin 2009, p.21.
[439] Haykin 2009, p.18.

successive computation affect the computation of previous elements, it is a feedback system. McCulloch and Pitts' units and nets are not incompatible with feedback connections. The other big distinction regarding the features of connections is the distinction between weighted and unweighted connections, that has been already clarified[440].

Concerning arrangement, relevant distinctions can be drawn according to (A) the set of the variable "time"; (B) distribution of units and (C) distribution of connections. (A) It is possible to conceive time as irrelevant, namely positing that the output of all units in the same layer is computed instantaneously, without any delay between them: Time can be posited in such a way that the computation in each unit in each layer takes exactly one unit of time. McCulloch and Pitts' nets are as such, and they are called *synchronous systems*. If these presuppositions are broken, thus it is posited that each unit computes its output independently and at statistically set times, then we have an *asynchronous system*, as for example the Hopfield networks. The former imply some complications in case of cycles, whereas the latter do not need specific changes[441]. (B-C) Distribution of units and connections can affect computation in relevant manners. The field is in constant growth, therefore it is complex to trace a list of standard models. The example I am going to provide exemplifies the difference. There is also an interesting chart provided by a member of a society for research in artificial intelligence. Since it is not a peer-reviewed publication, I just attach the link in the footnote. The examples from academic source are consistent with the graph, therefore I send to it even if it is not included in the references[442]. Regarding the distribution of units, Haykin explains that multilayered nets «[…] distinguishes itself by the presence of one or more *hidden layers*, whose computation nodes are correspondingly called *hidden neurons* or *hidden units*; the term "hidden" refers to the fact that this part of the neural network is not seen directly from either the input or output of the network. The function of hidden neurons is to intervene between the external input and the network output in some useful manner. By adding one or more hidden layers, the network is enabled to extract higher-order statistics from its input»[443]. Examples are provided in the document linked in the previous footnote, but it is important to fix these terms, thus I prefer to report them from an acknowledged source, even if they are "common sense" in the field.

In the end, a further criterion of classification could be drawn regarding the weighted ANNs

[440] Rojas 1996, p.46 include this feature as a relevant feature for classification too.
[441] Rojas 1996, p.46.
[442] Accessed in 26/02/2020: https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464.
[443] Haykin 2009, p.22.

which are capable of learning. In this subset, it is relevant to distinguish supervised and unsupervised learning. *Supervised learning* includes presentation of a large set of samples and an indication about the desired outcome, then the net operates its computation and compares the values obtained by its computation with the desired output and adjusts the weights according to another computation, proportional to the value of an *error signal*[444]. Briefly, that which happens is that datasets are prepared, where both relevant features and labels are provided in numerical terms. For example, if one wants a net that classifies samples of moths into different categories (target variables or response variables), a dataset is prepared with relevant features (predictor variables or independent variables) such as width of wings, pattern of texture on the body and size. These features are coded in such a way that values mirror differences and similarities. The net has a layer in which the correct answer is provided and is therefore coded in a compatible way with the data the machine is going to produce. At each presentation of the sample, the machine computes independently an output which is equivalent to a random guess, if the beginning bias and weights have been fixed at random. At each computation, a difference is calculated between the data stored in the layer that represents the correct answer and the output achieved independently. Weights and bias of units are adjusted accordingly.

In *unsupervised learning*, there is no pre-set desired outcome, the net needs to find target variables and response variables on its own and associate them with units sensible to differential patterns[445]. That which the system does is to examine statistically its mapping of the received input and produce independently a model of it, then it repeatedly compares the result with new samples, until the model produced matches successive samples. The possession of a model built over statistically significant distributions allow the machine to predict forthcoming samples, in both a statistical and a dynamical sense, i.e. respectively to elaborate new samples and elaborate new dynamics concerning a scenario. If both the procedures are applied, we have a *semi-supervised learning*[446].

Regarding learning, Rojas explains that, even if computationally powerful, MP nets have their issues: «the computing units are too similar to conventional logic gates and the network must be completely specified before it can be used. There are no free parameters which could be adjusted to suit different problems. Learning can only be implemented by modifying the connection pattern of the network and the thresholds of the units, but this is

[444] Haykin 2009, pp.34-36.
[445] Rojas 1996, p.78.
[446] Haykin 2009, p.44.

necessarily more complex than just adjusting numerical parameters»[447]. Learning is acquiring knowledge, and knowledge is provisionally considered by Haykin (which adopts the definition from other authors) to consist of acquiring information (in the usual sense) for interpreting, predicting and responding appropriately to the outside word[448]. According to Haykin, positing knowledge in ANNs relies on the following rules[449]:

1. «similar inputs (i.e., patterns drawn) from similar classes should usually produce similar representations inside the network, and should therefore be classified as belonging to the same class»;
2. «items to be categorized as separate classes should be given widely different representations in the network». This is the complementary statement of the previous rule. Not only the computation for the analysis of data must retrieve common pattern in the samples, but at the same time must allow to code explicitly the difference between inputs;
3. «if a particular feature is important, then there should be a large number of neurons involved in the representation of that item in the network». This is a biologically inspired principle. Since the detection structure of the brain is pyramidal, in the sense that a neuron at the top fires only if a certain pattern of previous neurons fire (thus, as in MP nets, the unit at the top states the information of the units below), a larger area for a certain feature allow a more fine-grained analysis, since more *specific conditions* can be coded, so that false positive are minimized and informative (low probable but statistically highly correlated with a positive outcome) features can be grasped;
4. «prior information and invariances should be built into the design of a neural network whenever they are available, so as to simplify the network design by its

[447] Rojas 1996, p.55.
[448] Haykin 2009, p.24. By the way, notice that this definition seems reasonable, but actually I think it is instead quite tricky, and if one is inclined to be suspicious, it could even be charged to be *ad hoc*. For this definition equates a general act (acquiring information) with the commonsense *practical consequences* of possessing knowledge. Thus, knowledge is ascribed on the basis of possible actions that follow from it, but it is unclear if there is a biconditional relation or not, which is demanded for the definition to be binding. I say one could think that is even an *ad hoc* definition because it ascribes knowledge by ascribing a coherent behavior, and since behavior is everything machines are able to achieve at the moment, this attribution is quite convenient if one must defend a concept as "artificial intelligence" in a strong sense. Of course, these remarks are not directed neither to Haykin, which is dealing with other topics in his book, nor to scientists that are trying to develop consistent solution to scientific problems. My remark is directed towards those who are tempted to ascribe knowledge in philosophical sense to computing machines on the basis of their capacity of acting accordingly to the environment, for two reasons: a risk of trivialization – since even a sensor acquire information, predict the risk and respond appropriately in a certain sense – and for the risk of insufficient connection between cause and effect in the explanation.
[449] Haykin 2009, pp.24-29.

not having to learn them». This principle prescribes to embody successful correlations and falsifications within the design of the network. Haykin points out that the more prior knowledge is built into a net, the more it becomes specialized: this improve performance and make it easier to manage, but at the same time constraints its utility.

1.4. The Role of Architecture and Mathematical Strategies in ANN

In this subparagraph, I will provide one example for ANN architecture (feed-forward perceptron) and the related algorithm of backpropagation for supervised learning. This example is provided so that the role of architecture (distribution of connections and computation of weighting) and mathematical properties in ANNs can be exemplified in a case more sophisticated than MP nets. In this way, the reader will be put in condition of having a basis for a concrete understanding of the key concepts of machine-learning in brain-like models of computation, as "knowledge", "learning", both "supervised" and "unsupervised", "categorizing" and similar introduced in this and the previous subparagraphs. This exam has the same purpose of the detailed account of the Von Neumann architecture in the previous chapter: providing a factual basis for understanding the theory of computation, understanding what lies behind human-like terminology as the one just mentioned and what room is left for ascribing representation to this kind of machines. I decided to give an account of just the Perceptron because more advanced examples imply a mathematics that I am not able to fully handle, but since this model, even if simple, provides a decent understanding of the basic issues, I decided to give a detailed explanation only for this case, and discuss informally some relevant examples for the other categories in the successive subparagraph.

Frank Rosenblatt's *Perceptron* in 1958, successively revised in 1960, is the first example of concrete ANN with weighted connections. Its computational properties were analyzed in a famous work by Minsky and Papert in 1969, *Perceptrons: An Introduction to Computational Geometry*, in which they generalized some aspects of Rosenblatt's model. *Perceptron* is a device for solving visual recognition tasks. Some relevant properties have a biological background, which I am going to indicate.

Perceptron[450] has a first layer that receives the sample, called "retina", which as an actual retina is a surface divided into pixels with binary states, that activates accordingly to the distribution of the input over the surface, and it is partially connected (i.e. the mapping is not one-to-one) to a further layer of elements called *predicates*, which transmits just a bit

[450] For the subsequent explanation, refer to Haykin 2009, pp.48-55.

over the input[451]. Predicates can be as computationally complex as desired, but their connections are fixed and unweighted. Predicates map the surface of the retina according to a *retinotopic* principle[452], namely they preserve the spatial relations between one layer and the next from retina up to the last layer. Each predicate is responsive to every stimulus that falls into a certain area – which thus can be more than one pixel – that can be defined "receptive field": the area that produces a firing if it is stimulated. The number of predicates that map the retina constitutes the *resolution* that characterizes the incoming stimulus compared to the stimulus in the retina, whereas the number of pixels form the resolution of the retina. The width of receptive fields and resolution of intermediate layers affect the possibility of solving problems regarding pattern detection and individuation. Rojas deals extensively with the topic[453].

Notice that the closer the relation of predicates is to a bijective relation, the higher the resolution is, and the stimulus is preserved in all its fine-grained details, but the same does not holds for successive layers: in this form, the stimulus is not informative in Shannon's sense, since it just states that there is something in general; in successive layers there must be some kind of differential response, namely a specialized response sensitive to specific and more improbable patterns of activations, which are therefore more informative.

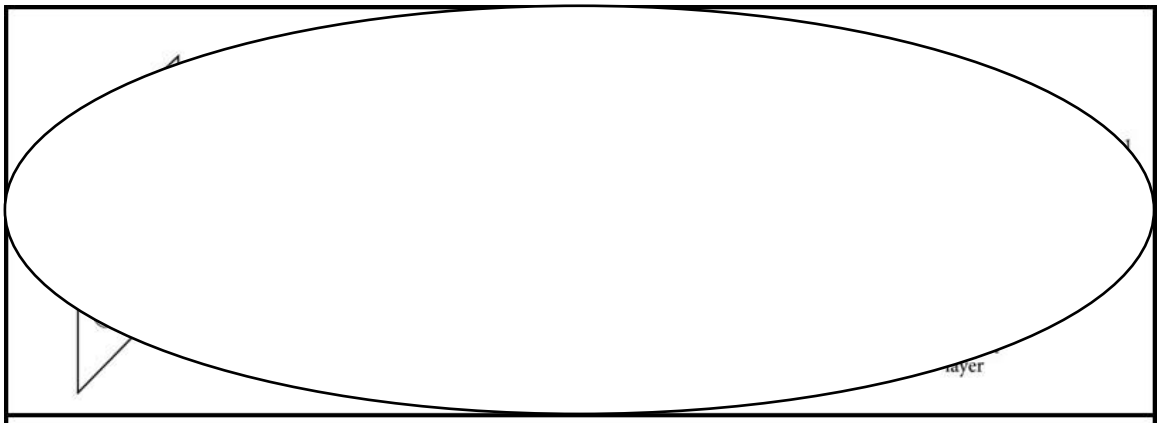

Fig.7. On the left, there is a schema of multiple properties and their weighted connections to an intermediate unit. The 'θ' designates the threshold value. On the right, a multilayer perceptron is showed, with the 'input layer' being the layer of properties. There can be as many property one wish to set, and they can be expressed by more than one unit. The figure on the left is in Rojas 1996, p.57.

Predicates send their output to threshold units – units whose function *f* acts accordingly to the structure that I reported in the previous subparagraph – and those units may be connected to further units or to an output layer, and implement the compatible strategies discussed in §1.3. Figure 7 may help the reader in understanding the structure of the

[451] Rojas 1996, p.55.
[452] Kandel et al. 2013, p.468. For the connection between biological retina and artificial retina, see Rojas 1996, pp.68-70. See also the successive pages of the book for more on constructing artificial retinas.
[453] Rojas 1996, pp.58-60.

perceptron and prepare him/her to the explanation of what kind of knowledge these systems possess. As the reader can appreciate in the figure, perceptron is a feedforward structure, and its units are weighted McCulloch and Pitts units. Here the connections for units computing error signals and weighting adjustment are omitted, so are also the bias inputs, which may be or not represented by external inputs or coefficients in the computation. I will introduce them later when backpropagation is discussed.

The problems compatible with perceptron are called problems of *linearly separable* classifications[454]. The reason for this name is explained by the analytical representation of these problems. I will not discuss the whole mathematics, since that would require to introduce mathematical, technical features that would expand unnecessarily the exposition.

Let for the sake of exposition have only two inputs, their weights, one output directly from the subsequent one unit (without any multiple output and no hidden layer) and two possible classes $\boldsymbol{C_1}$ and $\boldsymbol{C_2}$. In this case, the equation of the integration function has the form $\sum_{i=1}^{m} w_i x_i + b$. It is possible to define a *vector space*, namely a set whose elements can be added together or multiplied with real numbers according to certain axioms, which is not necessary to enquire in detail here. For example, the familiar three-dimensional space is a vector space. An *hyperplane* is a subspace or a translation of a subspace of a vector space having one dimension less. For example, a plane is an hyperplane in the tridimensional space, a line is an hyperplane of a plane, and so on. Not all hyperplanes are vector spaces.

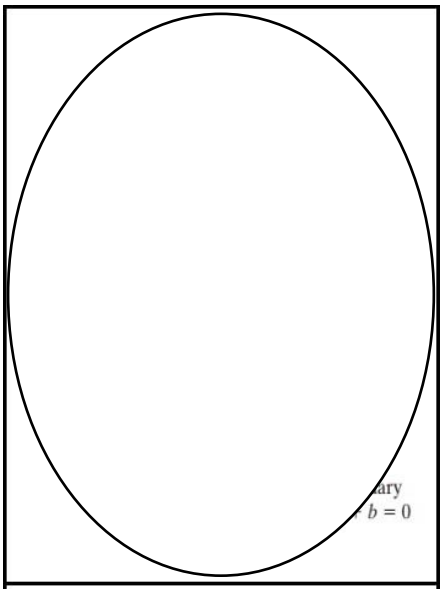

Fig.8. The hyperplane representing geometrically the decisions of the machine. The figure is in Haykin 2009, p.49. I don't possess any right for this image. All the rights belong to their owner in the original text.

The concept of hyperspace allows to deal with infinitely many input variables, but let me introduce a simple example. Since we set only two variables, the space has just two dimensions and two axes, as the ordinary cartesian plane. The equation of function g is simply $g = w_1 x_1 + w_2 x_2 + b = u$. The reader familiar with analytical geometry can see that this function can be equated to the equation of a line in the plane, namely $ay + bx + c = 0$; in fact, according to the threshold function that we have introduced through Rojas and Haykin in the previous paragraph, the value 0 marks the threshold between the

[454] Rojas 1996, p.63 gives a formal definition. See also Haykin 2009, p.48.

asserting/deasserting value. The binary value is obtained through the application of a *hard limiter* to the output function *f*. According to the premises posited so far, we have the situation in figure 8. The reader may see that the purpose of the net in terms of analytical geometry consists of defining a threshold that separates the distribution of data into a certain number of classes. Equation g sets now the so-called *decision boundary*, which is computed in each case by $f(g) = y$, and it is of course the value that determines where each element of the sample is assigned.

Since perceptrons are capable of learning, if *y* is not the desired outcome, it is possible to implement (automatically) a (feedforward) *backpropagation* algorithm: an algorithm that adjusts the weights according to a computational description of the difference between the desired outcome and the produced outcome. Haykin describes the strategy of learning by backpropagation as follows (I omit the equations, since they are not relevant for my purpose)[455]:

1. *initialization*: set the starting weights and biases at random, but some mathematical constraints must be respected in any case;
2. *presentation of training examples*: then many examples are provided to the retina, and for each of them the successive points are repeated;
3. *forward computation*: the calculation of the composed function $f(g(\mathrm{x_i}))$ for computing the output of the units and, in the end, the output of the output layer;
4. *backward computation*: see figure 9 for a functional schema. Once that the output has been computed, the output layer transmits its result to a further layer that computes *the error function*. Backpropagation uses strategies that aim to reduce the value of the error function up to a minimum or to 0[456]. This is always possible for perceptron, since the computation of their error function forms a convergent series[457]. There are also

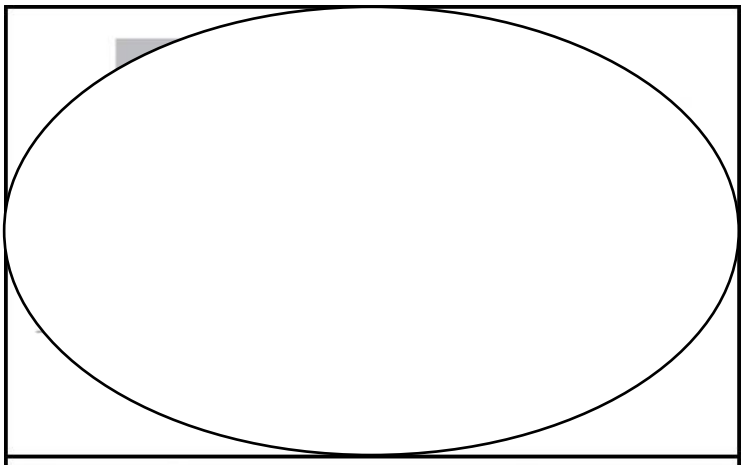

Fig.9. A multilayer perceptron with backpropagation and supervised learning.

[455] Haykin 2009, pp.140-141.
[456] See Rojas 1996, p.156.
[457] Rojas 1996, p.80 and following.

strategies for accelerating the reaching of this convergence. An example of calculation of error is the difference between the values of the output generated automatically and the correct output provided during supervised learning[458]. The result of the error function is then integrated in a computation whose output is included into the computation of the input layer, so that their weighting and bias values are adjusted according to this difference. This produce a strengthening in connections whose firing is statistically relevant for the individuation of correct cases in previous and future samples;

5. *iteration*: the error function goes gradually to the minimum or zero value, therefore until this happen the computation goes on many times.

The picture I gave, due to my lack of advanced mathematical and computer science expertise is very simple, but at least the general principles are correct, and I will ground my conclusions on them.

I hope I put the reader in condition of understanding the basic principles of ANNs: how the distribution of the structure of connections can affect computing in the case of perceptron, what is learning through the example of a supervised learning with backpropagation of error for weigh adjustment, what is knowledge, what is the role for them of weighting and error function. To sum up, knowledge turns out to be the distribution of weights and connections in such a way that they replace a declarative, stored knowledge of relevant features – the most useful stimuli as coded into activations in the input layer of properties – for classifying elements in future samples. Notice that it is not necessary that the equation is a line: other kind of linear separation are available. As in the case of classical architecture, the perspective of the machine is limited to doing the computation and in being in a certain state. This is of course relevant for the discussion in the subsequent paragraph.

§2. Do Neural Nets Compute?

2.1. Answer to the General Question Whether Artificial Nets Compute

The example of perceptron shows the relevance of the following features for the activity of a neural network[459]:

- *ANN architecture*: the distribution of connections allows or makes incompatible some strategies for learning[460], hence for activity we want these devices to perform,

[458] Rojas 1996, p.156 and following.
[459] See also Jain, Mao and Mohiuddin 1996, p.33.
[460] Jain, Mao and Mohiuddin 1996, pp.35 and 38 attempt a general classification that correlate architectures

such as classification, recognition, attribution of classes, data mining and so on. I discussed just the example of perceptron, but the online chart provides many alternatives, whereas the texts I referred to are mainly interested in the mathematical aspects of ANNs, even if they also mention some cases as Hebbian networks, radial basis networks for implementing kernel methods of clustering, and some other. The case of perceptron makes intuitive how a different distribution of units in the input layer and connections in subsequent layers may affect the procedures. The reader interested in biological examples may examine the pages on columnar organization of visual cortex V1 in Kandel et al. 2013. The principle is the same: the distribution of receptive fields and the differential response dependent on the connections produce sensibility to relevant property. In perceptron this principle is concrete, but more abstract options are available, dependent on the next point. The most relevant difference between classical architectures and connectionist architectures regarding the question of computation is that there is no separation between memory and executing unit: units are structured to be functions themselves, and signals are their arguments. For the sake of accuracy, if net is simulated, then units rely on a Von Neumann model of computation; if net is physically implemented, then not all the computation in units is hardwired: some of the models reviewed in Jain, Mao and Mohiuddin 1996 or in chapter eighteen of Rojas 1996 include the division between memory and execution. Actually, no computation is completely hardwired: a more or less large part can be carved and made physical and automated, but a full hardwiring has never been done, as far as I know;

- *mathematical description (conceptualization)* of relevant aspects for a task: in the case of supervised learning, the error function consists of a direct measure of a difference between the values in the output layer and the values in the layer that registers the desired outcome. Moreover, I was saying, distribution of connection and weighting are computed in a relatively straightforward manner in case of perceptron, but more complex and abstract options are available. For example, the radial basis function networks conceptualize their input layer as a multi-dimensional space and the inputs as series of vectors described as matrixes. The mathematical concept of the clustering they produce corresponds to a geometrical interpretation of the computation in which multiple points – corresponding to the

and procedures for learning.

classes we want the machine to define and fill – are individuated at random (K-points) and then they are moved in this space according to a vector representation of the error and the hidden units[461]. The mathematical construction of the learning algorithm and the mathematical properties and theorems are that which (1) explains the processes ongoing in the machine and (2) makes sense of how the operations can accomplish the results they are accomplishing.

I summarize the definition of computation again so that the following discussion can be followed more easily. The definition of computing I provided was structured as follows:

I. "computing" means to execute an effective procedure over formal terms according to logico-mathematical rules. It must be possible to define the effective procedure, but the concept leaves undefined *how* the procedure is implemented;

II. ascribing computation depends on two factors: necessity of describing an activity according to some computation-related formalism and a ground of this necessity such that it allows to attribute computation to the objective reality of the computing subject.

In the case of classical architectures, the necessity at stake was that of accounting for the properties and the development of the activity ongoing in the computational devices, and the ground was the structure of these devices, whose internal relations are explicitly designed according to logico-mathematical operations and purposes. In the case of connectionist architectures, necessity of referring to mathematical relations and properties is even stronger: as I was saying, the second most impacting factor in the activity of the ANNs is the mathematical description of features relevant for the task at stake: machine learning is possible only because a specific way of conceptualizing in mathematical terms inputs and solutions to problems of classification has been developed in advance. This holds for error functions, but also for vector spaces as representing the input layer, or radial basis functions as defining classes. Without this apparatus, it is impossible to explain how the weights change, how ANNs adapt their structure to set of samples and so on. The description would reduce to taking note of the changes, but this is insufficient for expressing the properties that regulate the phenomenon. The objective ground for ascribing computation is thus given: in simulated networks, unless one goes down to the hardware implementation and simply extend the conclusions of the previous chapter, networks consists of iterations of functions, this is that which they are at the abstract level of implementation; in hardware network, again we have components and operations that are

[461] See the detailed account in Haykin 2009, pp.230-260.

designed to be mathematically structured per se.

Mathematical nature of architectures and operations in ANNs are thus given; it remains the question whether these nets compute in the proper sense, namely if there is an effective procedure. I would like to offer a twofold argument for a positive answer, based on the informal, conceptual understanding of the case of ANNs. First, the flow of signals and the concrete, singular computation cannot be completely specified by an external observer, since the computation in the hidden layers and the adjustment of the weights is not fully explicit, nor they are the selection of features and the process that brings to selecting some features instead of another in unsupervised learning. Nevertheless, the *algorithm for learning is fully specified*, and so is its application: there is a mathematical description of the conceptualization that those devices implement and specify for each concrete case, and these mathematical descriptions are realized by following a progressive application of functions to previous functions. Amongst, the functions that regulate the process, some count as local rules localized in each node, others as global rules localized in the distribution of connections all along the net. The enquiry on the concept of computation allow to call consistently these procedures algorithms – as in fact both Rojas and Haykin do along the text – since even if their development is not fully describable, nevertheless it is possible to say that the operations inside the net are developing according to mathematical rules defined over a flow of signals. Second, ANNs consists of composition of functions, i.e. functions applied one over another, therefore if a function is an effective procedure, a composition of effective procedures is still an effective procedure (even if there is no warranty that the procedure reach its end).

Over than this general argument, I would like to add something more specific, related to current debate on the topic. The question whether neural nets compute is widely debated in philosophy, and the answer largely depends on how ANNs are conceptualized and how computation is conceived. Piccinini remarks that, first of all, if a debate between connectionist computation and classical computation has to make sense at all we must be in the position of distinguishing between computation and non-computation and, within the former, between classical and non-classical computation[462]. Once the general question has been set, Piccinini discusses a dilemma regarding computation, structured according to the literature on the topic: either connectionist architectures execute a program, or they do not compute[463]. The presupposition is that executing a program in some sense is the

[462] See Piccinini 2008, p.312.
[463] See Piccinini 2008, p.313.

necessary and sufficient condition for computing. This thesis has been defended by authors as Jerry Fodor and Zenon Pylyshyn.

For example, in their joined paper, *Connectionism and Cognitive Architecture* (1988), Fodor and Pylyshyn argued that the presence of a *symbolic* level is that which makes the difference. This is efficaciously summarized by the passage «[…] the Classical theory is committed not only to there being a system of physically instantiated symbols, but also to the claim that the physical properties onto which the structure of the symbols is mapped are the very properties that cause the system to behave as it does. In other words the physical counterparts of the symbols, and their structural properties, cause the system's behavior»[464]. Thus, Fodor and Pylyshyn maintain that the point is that rules that govern the behavior of the computing machines are concrete formal rules, that are defined over equally concrete symbols, where "concrete" has to be understood as "having a physical correspondent item".

Piccinini understands Fodor and Pylyshyn's position (that is documented also in other works) as said above, as "computing means to execute a program in the classical sense"[465]. Samuels 2019 defines along a similar line of though classical computation. I think Piccinini is right in giving this interpretation: a program is a computation over formal variables according to a formal language, and this is how the flow of signals and the transition of states *are described* in classical computing machines; they *are* equivalent to them. This account for both the legitimacy and the correctness of Fodor and Pylyshyn's thesis. Piccinini also considers a looser idea of biconditional relation between computing and executing a program, namely the thesis by Roth, according to which it is sufficient that a functional dependence can be specified for something to be said to compute, where "functional dependence" can still be understood as programs in Fodor and Pylyshyn's sense[466]. Both the options are denied by Piccinini, on the basis that classical computers are actually connectionist mechanisms too, since as such are the relevant parts that perform the logico-mathematical operations within the machine, and that it is possible to define a sense in which connectionist architectures compute, as in the case of McCulloch and Pitts' nets, but neither the former case nor the latter refers to symbolic manipulation as it is understood in the context of programming; context which we should follow, since it is the only available and reasonable common sense through which understanding "computing as

[464] Fodor and Pylyshyn 1988, p.12.
[465] Piccinini 2008, p.312.
[466] Piccinini 2008, p.313.

executing a program"[467]. In spite of this, in giving his own solution to the question Piccinini keeps the idea that some kind of symbolic element must be present in order to compute, even if it is not necessary that a program (in the sense there are programs in computers) is executed: those nets whose input-output relations can be mapped with a proper function over digital items compute, others do some operation slightly different[468]. This amounts to say that it is not necessary that there is a *stored program*, but that there must be an *input-output relation over symbolic items*. Programs, in fact, are actually functions that associate inputs to outputs.

I share the analysis of the arguments Piccinini does, and so the replies he gives, but I do not think this conclusion is correct. For as it has been shown, "computing" does not imply "program", whereas it is equivalent to "algorithm", which is in turn better defined as "effective procedure" according to the analysis I elaborated through the discussion of Soare's paper and the description of Turing Machines. Thus, it is not mandatory that there are symbols in order to compute; also because this would make things problematic, since properly speaking there are no symbols neither in classical computation, just significative signals, that become "significant" for the presence of specific structures and a presupposed compatibility between the programming language and the instruction set architecture.

Fodor and Pylyshyn's position is somehow sounding, since actually it is the sequence of signals that regulate the behavior of classical computing machine, insofar as they are input to whom instructions associate some actions, but at the same time I have shown that this is some metaphorical way of considering the matter. For instructions and data are distinguished according to their relation to the low-level architecture, so that the latter are translated into a series of control signals by a suitable decoder, whereas the former are processed through the ALU in a mechanical way because they have been decoded in such a way to be sent to the ALU through the registers. Thus, in classical architectures signals are processed qua signals, according to mechanical organization and compatibility – in this, Piccinini's replies is correct too – not qua symbols, therefore there is a conflict between the level of description and the level of existence that Fodor and Pylyshyn's account cannot reassemble. On the other hand, nor can Piccinini's account be suitable to accommodate the difference, since I showed in the example of perceptron and indicated in some examples of learning algorithm that computation of functions in Turing's sense[469] is performed in ANNs too, but there is no symbolic item in the classical sense: there is instead an

[467] Piccinini 2008, pp.313-314.
[468] Piccinini 2008, pp.314-318. Especially p.318.
[469] As accomplishing a calculation through an effective procedure.

information-processing procedure whose signals are equivalent to *elements* to be computed, *in the sense of inputs distinguished according to their material properties*[470].

Notice that the question of information is different in connectionist systems and in classical architectures. Concerning the latter, I showed that signals have an information value since those signals constitute a process of communication that affects the behavior of the components, and because information theory allows to formulate a reasonable sense for connecting the abstract content that those signals *extrinsically* represent and that which they are from the standpoint the machine may have on its own operations. In the case of the former, this is not the case: the input signals have a value of information that depends on how much they allow to reconstruct the distal object or the relevant features for accomplishing some task; internal signals are informative in Shannon's sense according to the structural and the mathematical conceptualization and properties of the operations of the net. For example, a signal in a very specialized hidden node of a multilayer perceptron is informative in Shannon's sense because its activation adds a relevant amount of precision regarding some aspect of the task at stake by signaling a feature that contributes much to the outcome of the process in the ANN; the activation of a hidden node that stands for a center in a K-mean clustering strategy is informative because specifies the belonging of an item to one desired category instead of some of the others. The reader may see that this is not the same context as before, and moreover this is a context that does not make even an abstract reference to symbols, extrinsic or not: *ANNs compute directly values while they calculate functions*. But this is the point: they do compute functions in the sense they compute an algorithm directed to a certain end through an effective procedure defined over values and implemented by structural features, otherwise it is impossible to understand what ANNs do. For all those reasons[471], I don't think that Piccinini's account of computation is correct in keeping the idea of symbolic manipulation in the definition: for an account to be developed consistent with the fact of computing machines both connectionist and serial, it is preferable to have a more pragmatical view on computation.

Of course, Turing's original *model* included a symbolic manipulation and a computation

[470] Symbols in the context of computation are symbols as understood in formal languages: sensible items that are manipulated according to their shape and a pre-given role in procedures. If this is the case, "symbols" must be items whose significance depends on the actions associated to their being received and elaborated through sensible and material aspects. Thus, "symbols" must not be understood necessarily as in human common experience, as for example "*", "a", "+".

[471] To summarize the point, they are: the mechanical basis of computation of functions common to classical and connectionist architectures (but differently structured in each); the necessity of ascribing mathematical operations to ANNs and the organization in functions of them too for their operations to be accountable, as in the case of classical devices; two senses of "processing information", both of which play a relevant role in ascribing computation to classical and connectionist devices.

that is better defined by Piccinini's thesis, but it does not seem to me, in the light of the previous research, that the concept is bounded to the model of TMs regarding the material execution or the nature of the arguments of the effective procedure. Something more abstract is possible and even necessary, for otherwise Fodor and Pylyshyn are right – there is no such a thing as computing without symbols – and in this case, nothing compute, because properly speaking there are no symbols even in the case of classical architectures. That which we are dealing with are signals that reproduce a formal language within a communication process of information linguistically structured, a communication process that guides a planned behavior, in the case of classical architectures; we are dealing with informative signaling that reconstructs an external state of affair and its description through structural properties and differential responsiveness in the case of connectionist architectures, which are the information handled by connectionist computing devices.

To sum up, in the light of the enquiry in this and in the previous chapter, I claim that artificial neural nets compute. The argument for this conclusion goes as follows:

1. *the definition of computation implies following an effective procedure defined over formal elements and according to logico-mathematical rules*;
2. *attribution of mathematical features and operations is mandatory for procedures and goals of ANNs to be explained*;
3. *independently of any stance we take towards ANNs, their behavior and activities depend on the properties at point 2, therefore they are objectively ascribed to ANNs*;
4. *significance of information processing as a conceptualization for describing the function of signals in computing devices can account for signals as formal items in both classical and non-classical computing devices*. The traditional picture of classical computation (Fodor and Pylyshyn 1988, Samuels 2019) – executing symbolic manipulation over concrete symbols qua symbols by following formal rules as such – does not fit the concrete case of classical computing devices and lead to deny computational activity to ANNs, so that their procedures become impossible to explain regarding their development, their rationale and their capacity to accomplish goals they actually accomplish. The same does any account that binds symbols qua symbols in an ordinary sense. Thus, it is better to introduce a wider notion of symbol, consistent with how formal languages define symbols but not implying meaningful symbols. A functional account of information in Shannon's sense, differentiated for classical and connectionist computation, can do

the job: in the case of classical architectures, we have seen that information theory provides an account of the *significance* of signals by relating the accuracy of transmission to the effectiveness of conditioning the course of an effective procedure, since information theory can describe the internal processes of computing machines in terms of probability of successive state-transitions; in the case of connectionist architectures, Shannon's information can account for the *significance* of signals by defining the differential response as an informative response, namely an event improbable enough to be a reliable ground for affirming something on another event;

5. *ANNs execute functions*, which are effective procedures, over signals whose significance consists of their being informative on relevant aspects of a certain task. In this case, information is not a matter of transmitting a formal language, but of achieving a net of differentiated responses that are constructed, manipulated and interpreted according to application and development of those functions and their composition. These differential responses themselves are the signs over which the computation is defined. Such an effective procedure is computed over sign-signals in informatic and semiotic sense. For this reason, I claim that computation can be ascribed to ANNs;
6. *Final remark*: ANNs do not have a stored program, nor they receive the rule, not even in some abstract sense. Despite this, ANNs embody both the procedure and its arguments: they exploit an identity theory based on interpretation and measure of information in order to implement the computation. Since this does not contradict in principle the concept of computing, I think it is possible to conclude safely that ANNs compute.

2.2. Consequences for the Question Whether Artificial Nets Represent

As in the previous case, the question is whether ANNs can implement *on their own* the specification of the referents of their operations and the rules according to which they operate. That this is not the case follows from the description provided in the chapter. ANNs are designed to manage a flow of signals in a behavioral way too, therefore, their perspective is limited to that which they are doing, without any access to a conceptualization of their own activity. Their responsiveness is implemented in structural and behavioral terms, without any access to the rule as such or to the interpretation of their operations, namely to the referent and the sense of it that their operations presuppose or develop, as in the case of their classical correspondents. In fact, their differential response

is based on the distribution of their connections and the process of adjustment they realize known as “machine learning”. They just execute this procedure, but are clueless regarding that which their referents are. Thus, the problem introduced in the §§6-7 of the previous chapter affects connectionist computation as well as classical computation, since both share the procedural nature of their dealing with (initially) rational operations and the behavioral conception of meaning proper of some semiotic symbolism.

Notice anyway that a connectionist architecture is closer to a representation. By differentiating its input and output in an explicit manner (as different arguments for different functions) there is a distinction between proximal and distal stimulus, and between their own description of the former through the latter and the proximal stimulus. This last distinction could be regarded as a kind of symbolic construction, if the two contents could be unified into a unique explicit datum whose parts are connected according to the relation of reference. These parts are instead connected by a causal relation mediated through mathematical properties, which cannot be equated to reference per se, at the cost of claiming that every system whose processes imply a physical separation between cause and effect can represent, which is clearly not true.

Chapter Fourth. Final Arguments and Conclusions

In the development of the discussion on the computational model of representation, I have being relying on three postulates: representation is a form of reference that implies replacement; the necessity of possessing the representation as an object composed of both referent and interpreted sign; therefore the necessity of being given of the representation to the subject as an explicit datum. In the next paragraph, I will argue for these points with as many arguments I formulated, and in the end I will show that they do not amount to the pointless demand that representation must presuppose consciousness, but to the impossibility of constructing the basic structure of representations with a behavioral and procedural strategy.

Once this has been done, in the second paragraph I will recollect the results of the historical survey on the relation between Behaviorism and Cognitivism, then I reconstruct the model of mind that Computationalism implies, so that the concluding analysis on the idea of "computational representation" can be developed. The second paragraph in this chapter *marks the beginning of the general conclusion of this work*. I am going to show that this model of mind implies a presupposition: namely, that mind is something that comes when everything has been done, as the legendary owl of Athena[472]. This presupposition is built directly into the explanatory model of Computationalism, because of the historical genesis that lies behind it. The crucial points of the model have already been deduced in the previous paragraphs, hence this further chapter just summarizes the previous achievements and makes clearer their consequences. The passage on the computational model of mind in general, contrasted to the computational model of representation, is necessary for a complete account of a computational representation to be developed, since only through the understanding of how Computationalism explains mind in general becomes possible to make sense of why something as a computational process over information can account for mental representations.

In the end, I will present a criticism to the computational explanation of phenomenal, mental representation, on the basis of the reconstruction of the model of mind implied by Computationalism. That closing paragraph argues for the legitimacy of a completely philosophical theory of representation, since it suggests that the material realization of

[472] It is a metaphor Hegel applies for making the case that something comes into play when everything has been done, i.e. too late for partaking in the development of the thing at issue. It is originally referred to philosophy regarding its analysis of historical facts.

thinking and the development of thinking are different things.

§1. Three Conditions for Being a Representation

1.1. Representation is a Species of Reference That Implies Replacement

This can be showed by some observations. I do not regard recurring to observations as problematic, since even if I am doing an effort for not presupposing the phenomenal concept of "representation", "representation" is a notion whose referent is empirically given and has its own history. Thus, it is neither possible nor legitimate to deduce it *a priori*. Moreover, in accordance with Ramsey, our interest lies in evaluating if a computational representation is a representation as this concept is conceived. Thus, it is indeed necessary to start from some basic observations. These observations are not limited to mental representation, since our interest is not in a straightforward equality of the two sense of "mental" and "computational" representation, but in understanding what it means in both cases to be a representation, in accordance with Ramsey's analysis.

The first point to be observed is:

Observation$_1$: whatever a representation is, it is a form of substitution.

Representation is commonly understood as *including* (*reporting*) the object it represents in some sense and yet representation remains distinct from it: all the relevant examples of representation acts accordingly. In fact, pictures literally include in their sensible arrangement the appearance of the object represented; allegories refer to it by some allusion, and so does allegorical symbols, as those in esoteric traditions, and they do so by *reporting* in their sensible appearance something that can be conceptually connected with their referent; compositions of letters include a meaning, in the sense they refer to it; equations refer to objects and states of affairs in the natural world, when they are properly interpreted; finitist formal calculus refers to mathematics, and this is why we are so interested in it. Representations *as objects* (representation-signs or representation-vehicles) stand for something else, distinct from the singular, material sign that can be indicated. Thus, it can be safely concluded that *representations as signs replace something else*.

There is just one exception: a representation can refer to itself as a content, not as an empirical sign. In this case, it does not replace anything properly speaking. This can provoke paradoxes, as those studied in philosophy of language[473].

[473] The paradox of the liar is a famous example. A famous paradox of reference of a sentence to itself as a

The second point to be observed is:

Observation$_2$: whatever a representation is, it is a form of reference in Frege's sense, i.e. representation is some kind of link.

The range of ways in which representations send to represented objects is variously articulated, and probably it can also be framed in different ways. Nevertheless, two extremities can be fixed: imitation and convention. I show that there must be a background knowledge for a sign to be interpreted and recognized as such, and that a sign must somehow provide a ground for sending to that which it stands for, or it would be a sign hard to be constructed as such.

Notice that the maximum grounding of the reference *on the sign itself is that the sign shows that which it stands for*: ostensive reproduction is the maximum form of inclusion a sign is capable of, since it expresses the idea that sign and referent stand one for the other because they are somehow identical without being numerically identical, and identity is the strongest form of inclusion, understood as agreement. Identity is the basic principle of ostensive replication. On the opposite hand, the minimum grounding of the reference on the sign itself is that *it has nothing in common with the reference, and it must therefore be associated with it extrinsically*: this is the case of convention. Since it expresses just this being linked by a relation that states the transition from one element to the other in an asymmetrical way, reference in general seems to be the better provisional form for conceptualizing the nexus of representing.

"Provisional" because of course the interest lies in *how* the two things are linked, but I am not going to develop this point in this enquiry. I will rest on the observation that representation can be conceptualized as a *species* of reference. I precise that it is just a species and not the whole genus because *the genus to which representations belong is the set "products of symbolical competence"*, in which there are multiple forms of reference, not all predicable of representations only. Notice in fact that there are many ways for constructing the replacement of a content by a material trace, and if representation must be something, it must be one or at least a clearly defined set of those modes of replacement. I will provide just some example to clarify the point, but I will make no attempt of determining further this relation. The reader must be informed that my working hypothesis at the moment is that *representation is only that substitution whose replacement of a referent through a sign implies (A) positing a sense* (in Frege's sense)*; (B) constructing an*

content is exposed in Cartwright 1971, and a famous proposal for its solution has been given in Kaplan 1973.

individuation grounded on a sign, by convention, ostension or a conceptual link. I started working on it, but there has been no room for a full development.

Anyway, let for the sake of argument be the case that "representing" means "replacing and sending to something else". Now I ask: does an alchemic symbol represents in the same sense a picture represent? Both stand for something else and send to it; does a book represent as a letter? the letter replaces a sound or an idea, and it sends to it through a social convention; does a finitist sign represent in the same sense of an arithmetical sign, of a portrait, of a semiotic sign? All of those replace something and send to it. It is evident from those examples that the relation of replacement and substitution belong to any kind of signs, and that there are multiple forms of signification (ways of exerting symbolic competence). Thus, a good theory of representation must account for all these different constructions when it speaks of signs, and I am going to show that a sign is always required.

1.2. Representation Demands a Sign to Be Given and a Background Knowledge to be Constructed

Even if the proof of these two parts of the second condition – necessity of sign and necessity of background, past knowledge – is based on the mutual implication of the two, it is possible to prove them separately, even if this amounts to following the same reasoning but in different directions. Despite this, each passage of the proof is indicated and separately argued. This period argues for the first premise:

Premise$_1$: the property of representing, whatever it is, lies in the relation of a competent subject with an appropriate object.

Why there must be a representation-vehicle? Replacement presupposes *leaving a trace*. Representation cannot be a removal without a proper witness of the act: if that which is represented is simply replaced *without any ground for reconstructing* the act of substitution, then the substitution causes just the asymmetrical (one way) *transition* of something to something else, but representation must be constructed in such a way that this transition can be *replicated*, and for this purpose a permanent trace of the act is necessary, i.e. the sign.

Regarding signs, as Tim Crane puts it, «one point that needs to be emphasized here is that pictures often need interpretation»[474]. This holds for every representation-vehicle, since no sign is self-evident per se. Crane explains the point with an example on an historical fact[475].

[474] Crane 2016, p.13.
[475] Crane 2016, pp.7-9.

In 1972 the NASA sent a space vehicle, called “Pioneer 10”. On a metal plate, reproduced in figure 10, several items were reported: a schema of an hydrogen atom (on the top), a schema of planets with their relative sizes (bottom), a woman and a man naked, with the man portrayed in a welcoming gesture and the schema of the Pioneer 10 itself in the background. Crane points out that any of these figures can be obviously interpreted by an intelligent being: the humans could be mistaken by chemical elements instead of living being, the atom could be mistaken for a realistic pictures instead of a schema, the diagram of the solar system in the middle could be mistaken for a living being.

Crane does not enquiry further why this could be the case, but it is clear that the point is that signs are not given with their own interpretation. The other intelligent being could have no experience of humans, or even humanoid form of life – as in sci-fi movies, its form could be that of the diagram of the solar system, so that he/she/it is more inclined to consider that as a form of life. The lack of a background knowledge and its necessity for understanding a sign as a sign is a common point, that anyone can experience: every time we meet a new sign for which we have no background knowledge and we are clueless of its context, then it is just a material trace on some material base for us. Sure, if someone tells “look at this and tell me what it is for you”, the hearer is naturally inclined to seek a meaning, but this is a context as well, and it presupposes the *general idea* of a sign-meaning relation and a consistent social interaction.

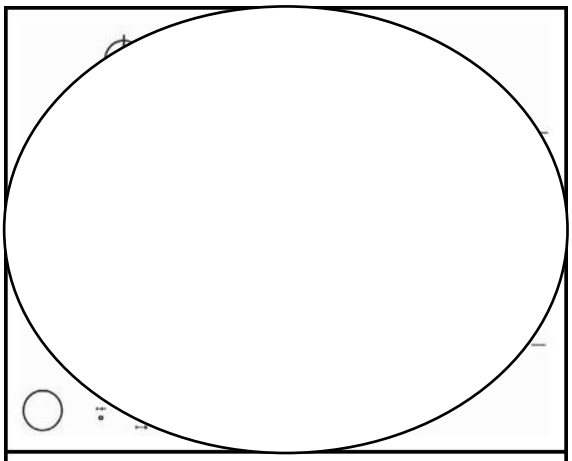

Fig.10. Reproduction of the incision on the metal plate on Pioneer 10.

Thus, a background knowledge to which a sign can be connected is necessary for a figure to be apprehended as such, as a representation. It also follows that *since every interpretation or symbolic production is the act of a subject, signs demand that a proper subject is in relation with the sign-object for it to be a representation*. In fact, let denies this is the case. If so, interpretation of figures as those in the picture should be interpreted – i.e. connected with their proper meaning – just for the fact of being observed. For if interpretation does not depend on the subject, it must depend on the object, and should be received when the sign is received. Subjects as humans can come in contact with signs only through sensibility, therefore signs should be interpreted just according to their sensible property. This is manifestly false, since a sign can be experienced correctly in terms of

sensible experience, but still be left without meaning: such a case should never occur, if interpretation were grounded *only* on sensibility. It follows from this that, in order to avoid a counterfactual consequence, it must be acknowledged that signs can be apprehended as such if and only if they are in relation (they are given in some sense) with a proper subject, where "proper" means "possessing an adequate background knowledge acquired in the past", as the case made by Crane exemplifies. The knowledge at stake must be possessed in advance because if this is not the case, then it is also possible that the subject does not recognize that something is a sign at all, i.e. not even that it is a sign in general, without attributing a specific meaning. It is easy to see that this conclusion holds for every kind of sign, not just for ostensive ones.

The object-signs must also be appropriate. Even if everything can be a sign according to the whim of a subject with symbolic competence, since it is always possible to associate subjectively everything with everything else by using a subjective convention, there are better or worst signs. In fact, if a sign does not offer directly (e.g. by ostension) or indirectly (e.g. by a common convention) a ground for the related term to be connected, then it remains a material object as any other, since we have just showed that every sign presupposes a background knowledge. Thus, "sign" is always something that offers a ground to a proper subject (in the above sense) for a content to be connected in some sense to it, once that it is in an appropriate relation with a subject. In the case of human beings, this relation can only be either being given through senses, or through internal state with the role of sign, as in the case of thoughts for humans, but in general every internal state that can be connected appropriately can be a sign. Let this be enough on "Premise$_1$" (P_1).

This passage argues for *the first condition for a representation to be given*:

First Conclusion: there can be no representation without the potential permanence of sign.

In the previous passage, I defended the premise that sign is a relative object, and I clarified that this depends on the necessity of possessing a background knowledge for a sign to be apprehended as a sign at all. In fact, sign is at the same time a material object, self-contained and independent from any relation the subject attaches to it, and a sign in the proper sense, namely a basis for the connection between itself and a further, different element, which is that which the background knowledge expresses. Let thus consider the following argument.

Suppose that the sign is removed. Insofar as it relates to the sign, background knowledge

contains its *identity*. In order to recognize Leibnitz in a portrait of Leibnitz, I must have memory of the relevant feature of his face; in order to see the meaning "lion" in this typewritten signs, I must remember not only the shape of each letter, but the content "lion" as well, at least discursively, as a nominal definition. Cassirer expresses the point as follows: «through the manner in which it is contemplated, the simple sensory material takes on a new and varied life. When the physical sound, distinguished as such only by pitch and intensity and quality, is formed into a word, it becomes an expression of the finest intellectual and emotional distinctions. What it immediately is, is thrust into the background by what it accomplishes with its mediation, by what it "means"»[476].

Thus, the removal of sign corresponds to the removal of something identified. Something that is identified as something else is by definition the subject of a predicate, namely "being something". If there is no subject for this predicate, then nor there is the relation of predication, and both subject and predicate return to be something independent, that stand on their own. In this specific case, background past knowledge returns to be *memory*, since it loses the role of being that which identifies. Its logical function has been changed with the loss of the sign. Since it has been shown that a representation is a relative status, depending on the conjunction of a material trace and a suitable background knowledge by a subject, the loss of even one of the related terms alters the logical status of both the related. In this case, background knowledge is no more the content of an identity (judgment regarding individuation of something as something else), but just memory of data. Thus, for there to be representation, it is necessary that sign is given.

Of course, it is contrary to experience that every time a representation is thought the whole relation is explicitly thought, nor all the parts each per se: as Cassirer notices in the quote, simply a different understanding of the sign is provided; the sign is given, but as if it were the content to which it sends. It follows from this that *for a representation to be given, it is necessary that in every moment the sign can be posited explicitly as something that is related to something else, that identifies it*. This is equivalent to say that sign must be potentially retrieved at any time and posited as "sign" (understood as a formal role in a relation) while a representation is conceived: *sign must be potentially permanent*.

A second conclusion follows from an analogous argument, which is *the second condition for a representation to be given*:

Second Conclusion: there can be no representation without the

[476] Cassirer 1923/1955, pp.93-94.

potential permanence of the background past knowledge that identifies the sign.

As the past background knowledge returns to be memory once that the loss of the sign happens and the relation of representation – whatever it is – is suppressed, in the same way the sign returns to be a simple object, independent from any relation that the subject attaches to it, once that the knowledge necessary for its interpretation is lost. For in this case the sign is just a material conglomeration or an internal state self-contained, therefore it does not stand for anything else than itself.

Notice that this does not exclude the hypothesis of sign as the content of itself. In fact, background knowledge needs not to be some declarative content, it can also be a procedural competence, as in the case of formal calculus. In this case, we have a sign, but as Cassirer points out, such a sign is somehow improper, because it only *presupposes* an interpretation without having any, at least regarding its composition and procedural treatment. In this case, it has not a meaning, but a significance for the subject that relates to it and acts accordingly.

Now that the two conclusions have been deduced, they can be assumed as premises for a third conclusion:

Third Conclusion: representation must be possessed as an integer object (datum), articulated into: interpreted sign-past background knowledge as explicit predicate that individuates the sign-referent.

This follows easily by the two conclusions and the Premise$_1$: it is just the composition of the three. I have shown that there cannot be representation without the potential permanence of the sign, nor there can be representation without the potential permanence of the past background knowledge or competence through which signs are interpreted. If representing is actually a form of reference, i.e. a relation based on the substitution of sign with something else, due to the mutual dependence of its parts and the necessity of them to be related for getting out from their condition of memory and trace respectively, then it results that representation as integer object, equipped with all its parts, must be given for it to exist. Since representations are substitutions, something must be substituted, hence a referent in general must be presupposed, even if it is not necessary that it is actually given[477].

[477] Cfr. Elgin 2010.

1.3. Representation as an Integer Object Must Be Given to a Subject as an Explicit Datum

This condition is the opposite of allowing implicit representations. I have already argued that it is necessary that all the parts of a representation are given together, since the economy of a representation is mutual, in this: each role presupposes the connection of the correlates, and a representation can last only until each of its parts is potentially present. Since representation is the result of the relation between a subject and a sign, then the permanence (potential availability) of the parts of the representation must be granted to the subject who owns the representation. Since this connection implies a replacement, but it needs to deny the dimension of succession in time for it to be different from a simple transition; since this is done by connecting the permanent sign to the faded referent through a link that, at first sight, consists of an identification of a content through another, the integer construct "representation" must be given to the subject as an explicit datum.

In fact, suppose this is not the case, namely that a representation is constructed implicitly, that is the subject who produces it is also unable to possess it in some form. The sign offers a ground for its connection with the past background knowledge, and vice versa, but nevertheless they are not merged in the representation: the latter becomes the *content* of the former, and the former the concrete existence of the latter (both when the sign is a state and when it is a material trace), but the two remains separate. "Being the content" is a shady notion, so let stipulate according to my working hypothesis that it means that some memory is the individuation of the sign, so that "being a content" becomes just the result of an operation of subsumption in a class. When one of the correlates is removed, each term returns to its original role: the sign turns back into an object, a material trace; the past background knowledge turns back to be a memory. Then, consider that the subject is the only medium that allows the connection: the ground in the sign is not necessarily a ground that the sign materially includes, as in the case of ostensive reference; the ground of connection can also be something that the sign allows, as for example being a basis for a conventional link, or the middle term that sends to a referent through a proper description, as in allegories, as in the case of Masonic symbolism; thus, a proper action of the subject must be performed over the sign.

If the result of this action is not given, namely if all the components of the representation are not posited as having the role they must have for they to be parts of a representation, then even if they are possessed at the same time they return to their original role, and the construction of the representation must be repeated indefinitely. For representation is a conservative relation: each part remains that which it is, it is only differently considered

one relatively to the other, therefore without their being related, even if they are possessed, they do not form a representation. If this is the case, then not only the parts, but also the relation must be possessed potentially as an explicit datum by the subject, as the middle term that connects each of the part and allows to retain the representation. In this case the relation must be given to the subject as an explicit datum, but without the correlates, this is just an empty formal structure for a forthcoming representation, not a representation. It follows from this that the representation as an integer object must be explicitly accessible to the subject, or it cannot be retained and must be reconstructed all the times from the beginning.

I remarked the impossibility of an "implicit representation", but there are internal states that are commonly conceptualized as representations, or at least as having reference, and that are not "given to a subject as an explicit datum" in the sense of being consciously accessed. It is the case of cortical representation of functions or anatomic parts[478], or as in the case of internal states that refers to external objects through an unconscious act of sight[479]. There is also the further case of unconscious thought with symbolic value in the sense of psychanalysis. Some clarification on this passage is necessary.

Above all, the reader must consider that my point here is always formal/structural, therefore the impossibility of constructing an "implicit representation" must be understood accordingly as: "it is not possible that some material trace or internal state can become a representation without the relevant relation and logical roles being posited *for the subject*". No bound is posited on *which form* the representation acquires for the subject (except for the case of knowledge, as I will show in the closing argument), hence in line of principle the idea of an unconscious state that is somehow constructed as a representation is compatible with my position. Nevertheless, representation must be distinguished from other relations of reference, and I do not think that this relation holds at least in the case of neural mapping, whereas the application of "representation" is at least problematic in the case of representational neglect.

Neural mapping is not representing: cortical areas are *hardwired in such a way* that their connection respects relevant relation between parts of the body, so that the differential activation of some areas corresponds to operational rules or relations (for example, my skin is mapped in a cortical area in such a way that I can distinguish where I have been touched), but there is neither need for this function to work, nor evidence, that the subject has a

[478] For the concept of "neural mapping", see Kandel et al. 2013, pp.343-344.
[479] For the concept of "representational neglect", see Kandel et al. 2013, pp.382-384.

representation *of the relation*. It is the same case of the top and bottom layers of an ANN: sure, there is a separation between the referent (external stimulus at the bottom layer) and its recognition (the outcome as expressed in the top layer according to the "categories" embodied in the intermediate layers), but the subject does not connect the two in any meaningful way, they *are materially related* through a concrete structure that implements a causal relation. Such an arrangement cannot be called "representation", since structure and causality are not a relation of reference; allowing such a use would probably have deflationary effects on the concept, and it does not account for common cases of reference.

Representational neglect is a more interesting case. Here the subject actually constructs an internal state with the value of representation, but it cannot access (read: be aware of) it. It seems also that he/she can perform complex evaluation over the referent of its representation[480]. Denying the value of representation of such cases would be inconsistent: either these are representations, or no internal perceptive state is, but this is false both as common use of the concept and according to my theory. It is also evident that there must be a causal connection between the uninterpreted sensible stimulus and the final perceptive state, or internal activity would be some kind of miracle or immaterial production, which is something inconsistent with our shared epistemic principles. Thus, representational neglect posits a twofold challenge: first, in which sense such representational states are "given to a subject as a whole explicit datum"; second, why should we not equate causal relation to at least some cases of reference? maybe I am still presupposing a phenomenal, intuitive concept of representation.

Starting by saying that the following replies concern that which for me is a perspective on which I am still reflecting, and on which dedicated enquiries should be spent, a reply to the first issue which is compatible with my theory is a Kantian perspective on the unity of self. The structural/formal perspective I adopted in dealing with the problem of representation does not compel to allow only conscious (aware) recollection of the parts of a representation and its synthesis into a whole item. My arguments just prescribe that the relations that compose a representation are posited, thus even if the internal organization of the subject constructs the unity of the representation by referring implicitly (namely without awareness) to an internal representation of itself (as for example the I-Think in the Kantian sense[481]) and by doing so it exploits such a term as a center for recollecting the

[480] See for example Cappelletti and Cipollotti 2006.

[481] For a brief overview of the concept and its function in the economy of Kant's theory of mind, see Allison 2004, pp.163-167. The Kantian "I-Think" is a function: it is an unconscious representation of the *unity* of experience possessed before any experience, whose duty is to be a formal and logical subject (and in fact its whole content is this being a unity and a subject) to which each possible sensation and internal content are

unity of the parts that form a representation, according to my theory the task of constructing a representation in the proper sense can still be fulfilled. Otherwise stated, my answer to the case of representational neglection consists of allowing that representation can be produced unconsciously, but still within the respect of all the conditions suggested by my results, by positing the ego as a *transcendental*[482] *function* over than as an empirical cognition, i.e. self-awareness. This is possible because the idea of representation I am defending is a matter of formal unity and logical relations, therefore there is no analytical reference to conscious (aware) experience and phenomenal *qualia*. For the reasons introduced in §3, I do not think that the distinction between representations implicitly constructed and aware representations affects the case of knowledge, which is a special kind of activity indeed, whose conditions *demand* the accessibility of the known content; thus, I regard the general possibility just discussed as not concerning the main purpose of the enquiry.

Regarding the second issue, the problem may be summarized as follows: if a causal production of logical relations and a complete natural explanation of mental phenomena must be allowed for epistemological issues to be avoided, why is it not possible to allow the application of the concept "representation" to cases of causal connection that link a state available to be conceptualized as "representation" with a referent, as in the case of internal states subject to representational neglect? In this case, my reply goes along an Husserlian line of reasoning, but at the price of confining the reply to the case of demonstrative knowledge, which is not a big issue for me, since my interest lies in philosophy. It just must be clear that my aim is not a general objection, but proving the reply that *at least in some cases, equating causality and reference in representation is wrong, and probably in general it is confusing and counterproductive for the epistemological reasons remarked by Ramsey*.

In his *Logical Investigations*, Husserl claims that a twofold confusion lies at the basis of psychologistic theories of logic: the equation between the act of reasoning with the content of reasoning and the confusion between logical principles as norms of reasoning and

ascribed, so that the continuity of the activity of the subject and the unity of its experience is ensured by stability and unity of this representation. I think there is room for positing such a function, but this is not something I can discuss here. The reader may be willing to accept this as my provisional answer to the problem at stake. I just point out that this hypothesis does not imply a vicious regression of the problem of representation: the "I-Think" does not work as a representation in fulfilling its role, it is instead a given content, since it is not constructed as something that refers beyond itself, but as a primitive datum.

[482] "Transcendental" in Kant's sense means "that which makes possible cognition of objects and experience before any concrete cognition takes place". Cfr. Kant 1787/1998, B25, p.149.

logical principles as causes of reasoning[483]. As a proof, Husserl stresses that the logical justification and the causal genesis of reasoning are not the same: interesting enough, the author makes the case of a computer, and he points out that if one wants to explain why such a logical device function in the way it does, he/she could not call for the logical meanings involved, but for its causal structure (actually, we have seen that this dichotomy is a bit less rigid, but not so much to quash Husserl's point), because this kind of device has not understanding or consideration of such a factor, therefore appealing to it in the objective description of the causal chain that regulates its behavior is illegitimate[484]. Husserl's point consists of remarking the impossibility of ascribing causal relevance to a factor that it is not even present within a process. He also adds a further argument amongst other. Husserl stresses that *truth is not a fact, i.e. if logical, mathematical or other theoretical laws were causal laws, they should presuppose that something defined in time and space is given, whereas this is not the case at all*[485]. Those two arguments show that the couples of elements equated by psychologistic theories of logic should be kept separated. For in fact when it is regarded as legitimate to equate the act with its content, then one is compelled to include the concrete production of a demonstration amongst the sources of justification of the demonstration, and this leads to confusing the acting with the logical grounding of its content[486], which are independent things[487]. By consequence, as Frege remarks «in proving the Pythagoras' theorem we should be reduced to allowing for phosphorus content of the human brain»[488]. The point of this provocative example is that *the causal process that generates a demonstration and the legitimacy of the demonstration are two things mutually independent, and as Husserl also emphasizes*[489], and as I would like to defend independently with the final argument in §3, *the latter is a matter of the content of principles and rules as they are in themselves, in their ideal value and discursive form*.

As a closing remark on the topic of unaware representations, allow me to clarify another possible objection. According to my thesis, the function that generates representations may

[483] Husserl 1920/2001, pp.48-51. See in particular p.49: «it seems that certain ready confusions have here opened the way to psychologistic errors. Logical laws have first been confused with the judments, in the sense acts of judgments, in which we may know them […]. With understandable ease a second confusion is added to the first: we confuse a law as *a term in causation with a law as the rule of causation*». Italics in the text.
[484] Husserl 1920/2001, p.50.
[485] Husserl 1920/2001, pp.54-55. See also his criticisms to Sigwart's theory at §29 and the remarks on this topic at §47 of the same book.
[486] See the case of the criticism to J.S. Mill at §26, namely Husserl 1920/2001, pp.58-59.
[487] Husserl 1920/2001, pp.101-103.
[488] Frege 1884/1960, p.XVIII.
[489] Husserl 1920/2001, pp.116-122.

well be unaware and implemented separately from the conscious (read: with awareness) access to internal contents, but if my arguments are correct, there must be a certain act of interpretation with the conditions I proposed, if a representation must be produced at all. Nevertheless, the forthcoming argument in §3 implies that knowledge is such a special case that the separation between awareness and formation of representation does not hold.

The case of knowledge is an exception because in that case *form and content of the representation are essentially unified*. As Cassirer puts it[490], the internal economy of the determined mode of symbolic production (mathematical understanding and the related symbolic production have mathematical laws as rules; logical understanding has logical principles and laws of inference, and so on) shapes the content of knowledge by providing both a certain mode of presentation and a certain form of relation between multiple contents. As I already argued through Husserl, since these formations of relations (namely the developments of the production of content within those defined symbolic productivities) presuppose the handling of conceptual, logical, mathematical and other form of principles *as such – and this for humans means to handle them consciously and in their discursive form* – for both their composition and justification, and (as I am going to show in §3) since it is not possible to change the form of knowledge without altering its content, then it is impossible to detach the formation of knowledge from its discursive composition and aware foundation, so that separating the aware access in the phenomenal consciousness from the formation of the content becomes an ungrounded division of two aspects of the same operation.

1.4. Consequences of the Three Conditions for a Computational Theory of Representation

As I have been showing in the previous two chapters, a computational mind stands in a procedural relation to its own objects both by definition and by construction: if "computational" in the definition must have a significance, it implies that this kind of intelligence operates by defining effective procedures over items that stand for or are managed according to a formal calculus. Only in abstract models, as for example McCulloch and Pitts' nets, these effective procedures can be executed independently from the succession of time. Nevertheless ANNs can consider their internal data only as arguments in procedures, therefore they must consider their signs only as physical inputs: when they become outputs that are not further inputs in feedforward of feedback circuits, they are emitted and pass outside any possible consideration of those machines. If I

[490] See Cassirer 1923/1955, pp.105-114, and for further clarifications on Cassirer's theory of signification as a modality of production of objectivity, see also Kaufmann 1949 and Solmitz 1949.

correctly argued the previous points, then any kind of representation (as well as any symbolic competence) is outside the possibilities of a computational mind in principle: for it can relate to its own objects only through the succession implied in the procedural relation it has with its internal data, but any retention constructed by substitution of a term with another and reference of the latter to the former (maybe) through identification is impossible, since every procedure is a transition from a step and its argument or arguments to another step with other argument or arguments.

This is the actual ground of Harnad's symbol grounding problem, as I was arguing in §7.4: simply because it is an operation that develops in time, every operation is bounded to be a transition and not a replacement that includes a retention. Physical retention as permanence of state as in artificial memory does not the job. For the kind of operation as representations to be achieved – cause of course constructing a representation is the act of a subject, and as such it is an operation developed in time too – it would be demanded to *store the whole relation* not as a bundle of signs. In fact, as Searle and Harnad showed convincingly, and as I have tried to argue too with the case of Finitism in mathematics, signs cannot be self-contained items: if they have to be appreciated beyond their aesthetical value or beyond their significance in the ordinary use of the term, signs must have a reference, hence for computational machineries to have self-contained symbolic activity, they should be able to specify reference on their own, but this seems not to be possible, since each computational operation consists in *replacing strings of signs with other signs of the same type*. It results from this that *it is impossible to construct a representation with an effective procedure on formal items*.

Notice that the problem is not that a phenomenal consciousness must be given: phenomenal consciousness constitutes the way humans deal with the problem of constructing a symbolic competence, but this does not mean that other options are impossible; the problem I point out consists instead of fulfilling some structural/formal conditions for something to be a representation. Neither the possibility of alternatives nor the demonstration of the conditions of possibility of representations in general solve the problem of describing *how* a representation arise in a system different from human consciousness: human symbolic competence is the only one of which we have a conception at least intuitive, since symbolic competence is a mental function, and we know only human intellect, and even this not entirely.

§2. On Behaviorism and Cognitivism, On Computational Mind and Computational Representation

2.1.The Historical Relation Between Behaviorism and Cognitivism

I showed in *Chapter First* that the authors who defined Behaviorism both conceptually and practically – Watson and Skinner – were mainly concerned with the epistemological status of psychology, and their historical context was the moment in which psychology was struggling for its acknowledgment as a scientific discipline. A struggle with many facets, to which none of the research programs developed within the field was indifferent. In this context, it is possible to see Behaviorism not as an anomaly, but as a step consistent with a pattern of reflection on epistemological questions, whose point of unity lies in individuating the conditions of possibility of a robust experimental practice, in which the scientific status of a field (psychology) seems to be grounded. The reflection of these authors can be summarized in three questions regarding mind: its observability; its availability as a variable that can be manipulated in experiments; its legitimacy as an effective cause, hence as an explanatory term.

Gardner portraits Behaviorism as entirely opposed to Cognitivism, and the authors who contributed to it as if they were extraneous to Behaviorism. Margaret Boden tells a more objective story, and depicts accurately the interaction between Behaviorist neuroscience and classical cognitivism – the classical cognitivism that Gardner summarizes in the conclusion of his book, and represented by the famous hexagonal diagram in the Sloan Foundation State of Art Report on Cognitive Science (1978). Following both Boden and Gardner and other sources in history of science, I argued that crucial ideas in the computational model of mind came from Behaviorist neuroscience, that for sure changed radically the model underlying Watson's and Skinner's theories, but at the same time this change was (A) still presupposing the same conception of science underlying the three questions Behaviorism asked to psychology, and (B) still reasoning on mind in *functional and procedural terms*. "Procedural" understood as "in terms of incoming stimuli that are managed for producing proper consequences", the only difference being the transition from overt behavior to internal regulation; "functional" in many senses: (B_1) *as in Functionalist Psychology*, where the goal was to individuate what consciousness and mind in general accomplish relatively to a naturalistic purposiveness (struggle for survival in case of James); (B_2) *in a mathematical sense*, related to the point of being procedural: mind is considered the *epiphenomenon* (see next subparagraph) of a manipulation of variables according to some quantitative or quantitatively describable process, Skinner was the first to coherently and explicitly formulate this kind of view, whereas Lashley and Hebb were the first to translate this conception into a neuroscientific research program.

The main ideas born in the context of Behaviorist neuroscience and developed by Computationalism within Cognitivism are the idea of living beings as self-regulating systems, whose internal processes are mathematically describable/mathematically regulated, together with the idea that organic activities can be exhaustively explained by the description of the underlying mechanism that realizes them[491]. Let me be sundry clear on this, in order to avoid misunderstandings: *I am neither telling nor even suggesting that existence and realization of mind and mental activity should be explained in some different way*. This would cause epistemological problems so widespread and radical that it is instead preferable that such a thing never happens. In pointing out this aspect, I am just summarizing that which follows from my historical analysis; instead, as I show discussing a further criticism to naturalistic conceptions of mathematical knowledge in §3, my main point against Computationalism as a philosophy of mind is that it does not even discuss the hypothesis that mind could not be a deceiving appearance, but a self-sufficient *state* that is necessary for certain activities to be accomplished *for a subject organized as human beings*. Thus, I am not in any way saying that mind is something unaccountable in naturalistic terms, I am just saying that there is the possibility that mind has been conceptualized wrongly (through a causal relation ground-phenomenon), and this conceptualization may be a misleading presupposition that should be explicitly questioned and examined, since it prevents to consider some relevant aspects of mental activity, as symbolic competence, cultural production, and also something that Aaron Sloman pointed out in a recent paper, i.e. the capacity of constructing explanatory framework and demonstrating their validity[492].

If my analysis is correct, there is a twofold continuity between Behaviorism and Computationalism: first, on the side of how science is conceived (as an experimental practice, centered on verifiability, manipulation of variables and reproducibility); second, on the conceptual side of considering mind as an effect of a process to which it does not contribute. There is instead a strong rupture with Behaviorism, in this: Cognitivism answers positively to the three questions whose negative answer was the core for the legitimation of Behaviorism. Thus, *the point of Cognitivism is providing that robust experimental practice that Behaviorism claimed impossible regarding mind and mental*

[491] See again the next paragraph for a detailed reflection on this point, additional to the survey of the sources in the discussion of Lashley and Sherrington; see also Von Neumann 1951.

[492] See Sloman 2018. Sloman points out an aspect that is relevant also regarding the topic of finitist mathematics. The author points out that Euclidean axioms, but also axioms in general, in human production of knowledge are *discoveries as any other statement in knowledge*. They are not mere presuppositions or pre-given assumptions. This is an argument that philosophical literature in philosophy of mind fails to discuss, but it should be taken seriously.

activity. The strong accuse by Lashley, «methodological behaviorism has all the faults of psychophysical parallelism plus that of intolerance»[493], is consistent with Cognitivists' view on traditional Behaviorism, but maybe cognitivist answer to that Behaviorism is not Cognitivism, but a more sophisticated Behaviorism, as in Lashley.

I am going to summarize what model of mind results from the previous enquiry, and I am going to show that it implies a strong identity theory between mind and brain, that allows a consistent reply to classical Behaviorism, but in turn shares its field of enquiry – behavior – and the fundamental presupposition of all scientific enquiries: that *describing the process of production (efficient causation) explains exhaustively all that the* explanandum *is and accomplishes*. In the case of mind, the content of every phenomenal mental representation. As a future research, it would be of interest enquiring *the conditions of possibility for applying legitimately this form of explanation*. Philosophy of science would make an important contribute in developing this enquiry.

2.2. The Computational Model of Mind, the Explanatory Value of Computational Mental Representation

During the enquiry on the concrete computational devices, two models have been presented. Jain et al. 1996 help to summarize the main differences between the two[494]:

1. *regarding processor*: classical architectures have a limited number of complex processors, with high speed; connectionist architectures have multiple, large number of processing units, with low speed which execute simple computing tasks;
2. *regarding memory*: classical architectures have memory separated from processors and localized; connectionist architectures have memory integrated into processors, distributed all along the net;
3. *regarding computing*: classical architectures have centralized processing in a sequential order, with the program explicitly coded into memory, but instructions are treated as data; connectionist architectures implement a distributed, parallel computation through the units, whose connection can be adjusted. There is no explicit program, but nevertheless they can be said to execute an algorithm collectively, whose aim is "learning", to which corresponds the progressive adjustment of weights in order to develop a differential responsivity consistent with a specific task they are trained to execute;
4. *regarding expertise*: classical architectures are competent in symbolic and

[493] Lashley 1923a, p.243.
[494] Jain et al. 1996, p.33.

numerical manipulation; connectionist architectures have good outcomes in classification and data mining problems;

5. *regarding mode of operating*: classical architectures need a well-defined and well-constrained procedure; connectionist architectures can manage poorly defined and unconstrained tasks.

These differences affect the models of mind that computational representations presuppose regarding the process that realizes them. Not only in the obvious sense that these are different models of processing information, but also because informativeness of processed signals assumes two different meaning regarding these two different models. In case of classical models, information is closer to its original purpose in Shannon, i.e. signals are informative in the sense they can be conceptualized through information theory as a concrete instantiation of the formal language they correspond to, which is communicated in such a way that the behavior of the machinery is affected in a desired manner, even if this does not solve the problem that machine code is different from that language, and that every formal language cannot be self-contained regarding the reasons of its being interesting, at least regarding the purpose of knowing. In the case of connectionist models, the processing of signals is informative because the differential response of the units become more and more informative in Shannon's sense the closer the processing gets to the output layer, but this model too suffers the same limitations.

I have argued that in both cases *the content of the information processing depends on* (1) *how information is processed* and (2) *the mathematical and structural properties of the computing subject*. Both the aspects are differentiated according to the list reported above. In turn, the details I spent for describing this modes of processing information had the purpose of showing that there are no such things as "understanding", discrimination", "semantics", "symbols" and "symbolic competence" in computational processes and devices in human sense: the referents of these words are *different per genus from their human counterparts*, so it remains the question of *how explanation is structured within Computationalism*, namely which is the answer to Ramsey's question "why "representing" of *x* should explain "representing" of *y*?". That which follows is a reflection on the line of principle, but I think it is sufficient for drawing important conclusions regarding a computational theory of mind as a computational theory of representation.

My effort has consisted of showing that *the order of the procedure, its organization (functional description according to algorithms) and the organization of the structure that realizes the computation are all that matter for an exhaustive explanation and description*

of contents (thus, of mental contents too) *in a theory of computation*. There is nothing over and above that which has been produced by, or according to, those factors in computational devices: once they are *described* into a proper form, nothing is left out. If this is the case, then *description is explanation, functional or not*. In the specific case of computational theory of mind, this is the form this precept assumes: *mind is the computational anatomophysiology of brain*.

For the sake of accuracy, notice that a computational theory of mind is not the naïve idea that describing a mechanism solves the problem, the point is that which the mechanism does *through its activity*. As Wiener remarked, «the mechanical brain does not secrete thought "as the liver does bile", as the earlier materialists claimed, nor does it put it out in the form of energy, as the muscle puts out its activity. Information is information, not matter or energy»[495]. In this passage, Wiener expresses the idea that processing information through a computation (let remind those are different things) is that which a computing device – brain included – does per se with its mechanical operating, and it is a product not reducible to its mechanical operating, it is not some kind of property or consequences, but the result of an activity. When humans speak, they accomplish a communication, they do not "secrete words as the liver secrete bile". The reader may now forgive me if I spent so much space in the previous paragraph for clarifying the issue of defining the condition for ascribing "computation" and be defined "computational": as I said, the premise of the semantic view of computation cannot be rejected, at the price of absurd consequences and of rejecting concrete matters of fact, i.e. we should allow computational devices to *objectively* perform a computation. Wiener said effectively that «no materialism which does not admit this can survive at the present day».

Despite of the emphasis on that which computational activities accomplish, the exam in the previous chapters was intended to be the most comfortable and (I hope) robust way for showing that computational activities (at least as they have been conceptualized and implemented so far) are inherently mechanical, non-semantic and entirely dependent on functional and structural constraints, and on properties and organization of an underlying structure; therefore, as Von Neumann himself remarks, the *description is the proper form of both exposing and explaining the activity, the result and the content realized by those computing subjects*[496]. This does not prevent this description to take into account the result of an activity, as Wiener remarks. Nevertheless, within this model of explanation, *the*

[495] Wiener 1961/1985, p.132.
[496] Remember the quote from Von Neumann 1951, p.24 in *Chapter Third*, §1.2.

efficient causation of a mental representation is the same as the content of a phenomenal mental representation: this is the core principle of how computational mental representation explains the phenomenal, mental representation. For in fact it is impossible to derive quality (phenomenal content and semantic relation) from quantity (structural and procedural feature of some mechanism) at the present state of our knowledge – but again, one day this derivation will have to take place, or proved to be just a necessary metaphysical assumption, even if not in the form pursued at the present days – hence at the moment the only viable option is to exploit the principle of sufficient reason: if some natural (or theoretical model of a natural) process at a (generally) lower level of organization (but more reliable from an epistemic standpoint) accounts for all that something else accomplishes by *producing it*, then the former explains the latter. This is a basic principle of reduction through causality that is discussed by Kim, with his argument of causal exclusion: it is just the complementary proposition of the principle of sufficient reason, and as we need to accept this in order to free empirical enquiry from useless and illegitimate *a priori* bounds, for the same reason we need to accept this principle. The fact that the description of the efficient causation/productive process corresponds to the account of the phenomenon to be explained is also a basic principle of scientific method[497].

Notice that this conclusion may involve a conflict with some theses that I had not the occasion to discuss, even if relevant. This is the case of the classical view of computation, as for example Fodor and Pylyshyn 1988 conceives it; under the assumption that knowledge is a form of representation, my arguments in §§5-7 of *Chapter Second* could affect Dretske 1981/1982's theory of information as possessing a semantic value; also the possibility of a non-reductive computationalism (thus a non-reductive computational functionalism) is something opposed by my theory.

At the end of this long journey, we are in condition to answer the three questions deduced from Ramsey 2006's remarks on the opportuneness of using the concept of "representation" for computational representation. The following list may constitute a summary of the enquiry:

- «How *x* represents? In other words, what does specifically (instead of "representing" in general) mean that *x* represents?»

  In this case, supposing the minimal notion of representing as including just the act

[497] Newton is the great absent of the paper, but I strongly recommend Moore 2003. The author shows that the relation between description and explanation is rooted in the history of Western knowledge from centuries: it is something whose genesis has been long and complex, and of which recent movements are no exception. See also Miller 1947 for the same historical derivation but from a more theoretical stance.

of replacing something with a sign that sends to it in some way, that is signs that have some referent (it is left undetermined if intrinsic or extrinsic, it does not matter in this moment), then reference of computational representation is divided into two cases: reference of internal effective procedures explicitly regulated by rules defined over the task at stake, and consisting of manipulating informative signals; reference of internal states that consist of distributed pattern of activation according to a differential responsivity to external state of affairs, and it is this coherence that makes such internal state informative. In the first case, reference is provided by the possibility for the effective procedure of mapping significant relations for the solution of the problem at stake, so that its development according to logical rule grants a consistent and truth-preserving treatment of the problem and a solution that can be regarded as true and legitimate under epistemological criteria of evaluation. The problem I addressed is that, even if one is willing to abstract from the fact that there are troubles in constructing a proper representation in such a behavioral manner, the issue remains that these systems can neither define nor access on their own both their referent and the conditions of validity of their own operations, so that no proper knowledge is possessed by those devices. This is not a problem if one just exploits them for solving problems or doing things, but positing them as explaining mental operation is another story, since humans can do exactly that which is left behind by computational processes, as Sloman 2018 points out, and as should admit also whoever share a notion of demonstrative knowledge even vaguely similar to the one I provided at the beginning. In the case of connectionist representation, the situation is analog. Connectionist representations warrant their reference through a causal/structural link between their input layer and the output layer, whose organization preserves a mapping that can even be independently formulated. Moreover, in the case of connectionist architecture we witness some separation between proximal stimulus, theoretical appraisal of the stimulus and response referred to it but, as in the previous case, the pure behavioral nature of this procedure, together with the impossibility of defining referent and principles of the operations, prevents to legitimately ascribing the possess of any representation/symbolic competence;

- «Why “representing” of *x* should explain “representing” of *y*?»
  According to my analysis, the crucial explanatory core consists of equating mental representations in phenomenal sense to the functional, procedural, organizational and structural features of the subject that realize operations we are compelled to

define as computational for them to be accounted. This does not amount to the absurd pretense of equating the *phenomenal* features of mind and mental activity to computational activities and procedures, but of equating the causal efficacy and the total effect "mental activity" has to be explained to some computational process and its products. I think this is the only consistent account of the rationale of this explanatory proposal. More will be said in the next paragraph regarding Computationalism and identity theory, but the problem is mainly that there is a presupposition implied by the principle of the explanation (equating efficient causation to mind and mental content): mind seems to be something that reaches the stage when the show is finished. This is in fact a thesis that Eliminativists and some reductionists explicitly state;

- «What is the nexus signified by "to represent" which belongs to both?»
  The nexus of "representation" should be further enquired, and it should be constructed in some way which is neutral regarding human consciousness, or any attempt of using it as a canon for evaluating the proper use of the concept would beg the question towards Computationalism and non-ordinary explanations. So far as my enquiry could be developed, I attempted to show that a computational representation is not a representation, since it lacks some formal features of representation because of the behavioral and finitist nature of computational procedures. This lack suggests in my opinion that any attempt of ascribing a form of symbolic competence to any computational subject is self-contradictory. Either another model of computation is produced, or the concept of representation is revised in some way which is not *ad hoc*. At the present stage, according to the outcome of my enquiry, there is no ground for conceptualizing computational states or operations as "representation", when those states and processes are considered only relatively to the subject who owes and produces them, i.e. under the assumption that a representation is not an "absolute object" in the sense previously discussed.

§3. A Fregean Reply to Explaining Intensional Reference Through Computational Representation

3.1. Recalling Frege's Relevant Remark

In *The Foundations of Arithmetic*, Frege discussed several theories of number. He reports that Stricker claimed numbers can be understood as muscular sensations[498], and Frege

[498] Frege 1884/1960, p.XVII.

objected that mathematicians cannot recognize their objects in such a definition, nor they can handle any mathematical proposition formulated in this way. The problem I want to address goes along a similar line of thought. Phenomenal, mental representations refer to outside states of affairs by providing an *intension* that makes them an object suitable to be understood and defined by some discursive content. Under the hypothesis "human mind constructs a reference to the outer reality through the definition of the referent of its own operations by a discursive content, which is the content of an intention (i.e. sense according to the Fregean use of the term)", as a closing step of the dissertation I would like to present an argument concerning the possibility of explaining this *intensional reference* through the model of computational representation. The point of this argument is similar to Frege's remark to Stricker in this: in both cases, the problem is that subjects organized as we are cannot handle the kind of items suggested by reductionist theories for the ordinary uses these items normally concern; and once this is noticed, it can be pointed out that every procedural/mechanistic explanation of mental operation neglects a fundamental aspect of human, mental operations, at least those concerning knowledge.

For the sake of accuracy, let be clear that it is not exactly Frege's problem, since Frege criticizes of Stricker's thesis that it is too tight to individual sensations, and all such a theory accomplishes is to make counterfactually vague the rigorous deductions and definitions of mathematics, and to mistake the signs with the objects, since all that is psychologically associated to numbers is their correspondent symbols. Moreover, such a view mistakes the awareness of a proof for the proof itself[499]. Frege's concern in these pages is therefore the necessity of separating mathematics from any attempt of a psychological foundation.

My point is instead gnoseological: maybe a science as the one implied by a computational model of mind would not fit the structure of *a science for human beings*, so that it would be of limited utility for its development, even if crucial for the *science of existence of human mind*. The reason for this last statement about "a science for human beings" lies in the following argument, and something has been already said in the closing remark of §1.3 of this chapter: the problem I stress is that there are good reasons for considering *knowledge formation and, above all, knowledge justification – which is the distinctive feature of knowledge from other cognitive states – as essentially tied to the discursive form in which knowledge is apprehended both in its content and its principles*. The two exergues at the beginning of this work attempt to summarize this point:

- *Cicero's quote* means (my translation) "in fact, in every matter the speech consists

[499] Frege 1884/1960, pp.XVII-XVIII.

of both words and objects, and neither words can have a place, if objects are removed, nor objects can have clarity, when words are removed". Otherwise stated, it means that there is an order in the speech that must mirror the internal relations of the referent of speech, but at the same time it is the composition of speech that gives a defined content to objects. When this is understood in the proper sense I am going to defend, then the objection in italics just reported above follows;

- *Minsky's quote* stresses that, in the special case of mind, we want the computational model to explain abstract products and contents of human mind, not only to reproduce faithfully the underlying mechanism, and since this abstract productivity and production is a part of the *explanandum* as the mechanism itself is (recalls Wiener's remark on this), then a lack of correspondence with the abstract contents and features of the *explanandum* is an objection to the physical aspects of the model too.

Now I move to the exposition of the final argument of the dissertation in order to recollect orderly those final suggestions.

3.2. First Informal Exposition of My Objection

Consider the following example. Let for the sake of argument suppose that a complete computational neuroscience is developed, and a proper translation of mental contents is achieved. In this case, probably knowledge will be equated to a certain internal state of the brain, resulting from some effective procedure mathematically regulated and developed. In this case, for any phenomenal content, it is possible to define the brain event that realizes it (I am not presupposing necessarily some excessively straightforward identification as McCulloch and Pitts), and to consider it entirely as a brain state. Suppose that this brain state instantiates some knowledge: it describes something according to a mode of presentation – an equation, a discursive description, a formal calculus or any other form of *sense* – in such a way that this sense can be objectively ascribed to the referent.

Now I ask: *what is that which constitutes sense related to the referent of some knowledge?* According to the computational theory of mind, once I described the underlying process that realizes that mental state, I account not only for its production, but for it being that which it is too, content included. Abstracting from problems regarding causal determinism, this means that sense is given in the form of the computational description of the underlying brain state. This conclusion does not demand any generalized identity theory: just the principle of sufficient reason, namely that this description includes all the effects that are usually experienced as a phenomenal state. Given the premises, I think the correct answer

to the beginning question is the following.

The *referent* of the computational state of brain is the state of the brain, hence a computational description of this state is a sense for *this* reference. Thus, that which this description presents is brain, and *only insofar as it includes the discursive sense, through which a subject refers to its objects, the computational description also concerns the referent of some of our knowledge*.

For example, consider the equation of the law of universal gravitation in classical mechanics. I choose some knowledge in symbolic form, so that it is ruled out any presupposition about the linguistic meaning potentially unfavorable to a computational theory of mind. That equation posits as a sense regarding the attraction of planet the direct relation of their masses over the inverse relation with the distance of those masses. The sense is the predicate "is the direct relation…", the subject and the referent of this knowledge is the phenomenon of "attraction". The content of knowledge here is the sense: the mathematical proportions expressed by the law. In fact, without that mathematical content, all that can be said about "attraction" is the sensible experience of the referent (objects fall to the ground), therefore it is such a mode of presentation that which defines (characterizes) the referent and articulates some further knowledge about it.

These remarks allow to notice that the content of knowledge must be kept identical in an explanation: transformations into extensionally equivalents or truth-preserving equivalents are not sufficient for preserving the content of knowledge, because in those cases knowledge could acquire *another sense, hence another content*. It is probably possible to construct many theoretical frameworks that are extensionally equivalent and such that ascribe truth and falsity in the same way as Newton's does, but they are *another sense*, another mode of presenting and therefore understanding the same phenomenon.

If this is the case, then a computational theory of mental representation, insofar as knowledge can be considered a form of representation, contradicts itself. Computationalism is supposed to provide a reductionist explanation of mind, one that rules out as unneeded the discursive, phenomenal form of knowledge. According to the previous remark, it is instead exactly the possession of *this* content *as such*, namely "$F = G\frac{m_1 * m_2}{r^2}$", that which both must be explained, and that which grants the possibility of demonstrative reference to an external phenomenon through a proper representation of it. Now I am going to show that any attempt of reducing this content implies a contradiction.

The contradiction at issue lies in a problem discussed by Jaegwon Kim in his *Physicalism,*

Or Something Near Enough. Kim explains that a proper reduction must expand neither the ontology nor the set of axioms of the theory that reduces. The second aspect is solved by stipulating suitable definitions, so that there is no need of using covering laws that would become new axioms of the theory that reduces. The first can be solved by a complete replacement of the items in the reduced theory by items in the theory that reduces: the former must not appear in the second[500]. This is instead exactly the case of any attempt of reduction of knowledge. For when the content "$F = G\frac{m_1 * m_2}{r^2}$" is reduced:

- either such content is posited as identical with the brain state computationally described, but in this case, it must be *exactly this content as reported in its descriptive form*[501], since otherwise it would be another sense, hence another content of knowledge. If this is the case, by consequence the reduction fails, because the content to be reduced reappears as such in the reduced form;
- or it is not, namely reductionist explanation transforms the mode of presentation of the referent in a pure computational description of the brain state, but in this case the referent of the new sense – the computational description – *is just the brain state*, as it was noticed above. Only to the extent the brain state is *also* the content of knowledge in its discursive form, the brain state is the embodiment of the above-mentioned knowledge of the event "attraction" and refers to it, but if this is the case, we have the same failure defined at the previous point.

Thus, a reductionist explanation of acts of knowledge is self-contradictory.

To sum up, the leverage of the argument is the conservative inclusion of the content that must be explained in the explanation, and the dilemma it generates for the computationalist explanation insofar as it must be a reductionist explanation too. This step of the argument sends back to the reference I made earlier to Husserl and Frege regarding the second branch of replies to the problem raised by the case of representational neglect as implicit representations.

The leverage of the argument suggests the same problem remarked by Husserl, and it is this aspect that clarifies furtherly what I mean when I say that Computationalism suffers a limitation similar to the one Frege objected to Stricker's theory. If the content of knowledge

[500] Kim 2005, pp.101-108.

[501] The passage in italics characters is not supposed to be interpreted as "it must be exactly this content as it is consciously given to a subject"; read instead as "it must be exactly this content as it is composed". The point here lies in preserving the sense, but since this sense is made of mathematical characters and relations, then the very same content must be given again in the reduced form – i.e. as that mathematical content.

depends on the form of presentation[502], as the argument I presented attempts to prove, then the discursive form in which humans compose sense and in which they necessarily apprehend the content of knowledge cannot be legitimately posited as irrelevant for the development of knowledge, since if this is the case, as it could be expressed in Cassirer's terms, humans are beings organized in such a way that *knowledge progresses according to the exertion of a particular shaping – mathematical rules of relation, logical rules of inference, conceptual forms of connection* and so on – *that is a special case of their more general symbolic capacity*, therefore knowledge development and knowledge existence are not mutually replaceable, and the moment of awareness is also the moment in which knowledge is both *composed and made actual knowledge*, not merely "accessed", as sometimes has been claimed. For if sense cannot be detached from the positing of knowledge, and if every shaping has its own way of being a "valid" and "lawful" (accordance with rules in general), then consciousness is also the "environment" within which knowledge can be both developed and justified, since there it is where content of knowledge is shaped.

Given this last conclusion, my argument points in the direction indicated by Husserl's line of reasoning in this: if the discursive form structures the content of knowledge and if it develops according to specific models of symbolic productions with their internal economy, then this internal economy must be set free to develop its own exertion *in the form in which it is experienced, and according to the ideal and conceptual reliability of the contents that are advocated during the process of justification*, i.e. for example logical rules exerts their regulative role *as logical contents as such, and the knowledge produced through them must be evaluated according to their ideal value*, as Husserl claimed in his work I quoted.

3.3. Second Rigorous Exposition

The argument can be reconstructed in all its conditions and steps as follows:

- *Condition 1*: knowledge is a form of representation;
- *Condition 2*: knowledge is demonstrative: it consists in describing objectively external reality, locally (as descriptions of individual processes like in biology) or systematically (as nomological description of wide processes or systems, as for example in physics), and in such a way that allows to provide reasons for its being

[502] It will never be stressed enough that the *form of presentation* – the declarative construction of content – is not analytically identical with the *mode of presentation* or *mode of experience* – the format of self-consciousness characterized by *qualia* aspects. In human beings they come together, but they must be at least conceptually distinct.

objective. Demonstrative knowledge is constructed through a form of presentation (sense) of some object (referent) in such a way that it fulfills these requirements;

- *Condition 3*: Computational explanations of phenomenal, mental representations consists of reductive explanation, achieved by translating the latter into a computational description of a relevant process and/or a computational state of a computational subject. It is an explanation such as it includes not only the material events, but also that which those events accomplish;
- *Condition 4*: phenomenal mental representations are also formed by a mode of presentation and the connection as reference to something different from themselves, namely a referent;
- *Condition 5*: Kim's constraints on reductive explanation;
- *Premise 1*: let there be an exhaustive description in computational terms of a mental representation. It is achieved through proper definitions, so that there is no breaking of the constraint regarding the epistemic expansion of the theory that reduces, and this is supposed to ensure that also the other constraints indicated by Kim are respected;
- *Premise 2*: such a description would result in some kind of formal description of either the state or the process realized by a computational device, which corresponds to the phenomenal content to be explained. For example, in the case of previous nets, the formal description may correspond to a number in binary calculus, to a formal proposition, or even to some graph. This is left unspecified: examples can be selected at will. It is sufficient that a different form is assumed and that it is consistent with a computational model of explanation;
- *Premise 3*: let for the sake of exposition suppose that the computational model at issue is a symbolic one, and that is implemented by a connectionist architecture. Let also give it an advantage: it can print symbols and store them in the same form as we see them once they are printed, so that it actually possesses a notation for its internal state. Since I showed that computational explanations exploit an identity theory, the explanatory system will print its internal state for every representation, once prompted to provide an *explanans* for a given *explanandum*. For example, it will have a form as: being a k digit number every possible content placed into a decision space linearly divisible, the representation in question will be something like $w_1x_1, \ldots, w_kx_k$;
- *Premise 4*: what is the referent of "$w_1x_1, \ldots, w_kx_k$"? Abstracting from the fact that

the computational subject cannot answer at all, this is the notation of its internal state, hence the referent of this is the internal state, and its sense is therefore a form of presenting this state;

- *Premise 5*: the notation should include the reference to the phenomenal, representational content to be explained by a computational description, and because of the hypothesis adopted (I refer to Premise 1) this is always possible by deriving the two according to some rule or empirical observation. Thus, it is possible to define the association T such that "$T(w_1x_1, \ldots, w_kx_k) = representation_{ph}$". The notation of the internal state may also include in its notation itself the phenomenal representation, without need of reporting the phenomenal form with an equality sign, since it is not obvious that such an advanced computational explanation has the same structure of the one in use nowadays, as Von Neumann remarks. Nevertheless, the association T is its *interpretation*, that which it stands for, whatever is the outcome of the reduction, since this interpretation depends on the premises and the conditions posited (as explained in §2.2 of this chapter, hence this premise depends on Conditions 3 and 5). This shows that any attempt of solving the issue by positing a mapping of the phenomenal representation by the computational state is useless: *the distribution of reference I observe in the next premise will be unaltered*;
- *Premise 6*: it is now visible that only the right side of the equation refers to the referent of the phenomenal representation, whereas the other side still refers to the underlying substrate or process that realizes the phenomenal representation. It refers to that which is known only *insofar as it is the phenomenal representation*, and the content of knowledge it brings is not changed, it is still *the one of the phenomenal representation*. One can replace whatever in the expression for identity: precisely because they are identical, the content of the phenomenal representation must collapse into the description of the internal state, and become its new content, otherwise it must be one of the two: either the explanatory left side has its own content, but this causes the computational state/process to lose the reference to that which is known and to alter the content of knowledge (sense as defined by Frege); or it has the same content of the phenomenal representation (the same sense, again understood as in Frege). If this is the case, some knowledge is accounted by *the content on the right side, not by the other* (left side), therefore not only the reductionist attempt contradicts itself, but we witness the outcome that the *reducens* is reduced to that which was supposed to be reduced! In fact, from the

standpoint of knowledge content and justification the computational explanation matters only insofar as it is identical with the phenomenal content;

- *Conclusion*: a consistent reduction of knowledge in computational terms is self-contradictory, even if one abstracts from all the issues I have been discussing so far.

I call this and variations of this argument *arguments for showing the duplication of Concept*. I write "Concept" with the capital letter just as an abbreviation for "demonstrative knowledge", which I regard as conceptually structured. The argument is called a proof of the "duplication of concept" because the content of the phenomenal mental representation proves to be duplicated in the case of knowledge by reductionist attempts of explanation.

3.4. Proofs of Conditions and Clarifications on Premises

The conditions can be easily defended. *Condition 1* is the idea that makes sense of the argument: if knowledge is not a representation, that is nonsensical discussing something regarding knowledge through the explanation of phenomenal representation by computational representation. It can be easily shown that knowledge is always a representation in the case of human beings: suppose it is not, then this means that every object of knowledge *is directly given to our intellect*, without any mediation by our internal states. Let for the moment abstract from whether or not phenomenal mental representation is epiphenomenal. In any case, either that which is manipulated by our intellect is the object itself, or it is not. If it is not, and it cannot be according to every evidence, then either that which is manipulated by our intellect refers to this object and replaces it, or not. In the second case, our mind is solipsistic, but no one wants to charge him/herself with this consequence, and arguments have also been provided that reject the conclusion in the history of philosophy. Thus, *internal states must be representations*, the most primitive ones. *Condition 4* easily follows from this: if things are not as they appear, as Kant already showed their appearance even if objective corresponds to a specific mode of presentation of their referent, hence internal states must be representation that can be understood according to the Fregean division between sense and referent. *Condition 3* is deduced in §2.2, so there is no need of further arguments; *Condition 5* is instead defended already by Kim: I can simply buy his argument. *Condition 2* is the only problem: it is a definition of knowledge, something that is widely debated. I can only ask to postulate this definition, since a complete argument is not possible here. It adds the objectivity of reference, but its being a representation can be derived by the fact that every knowledge is always mediated by an internal content, which is not problematic given *Condition 1*.

Premise 1 is stipulated by hypothesis for the argument to be articulated, whereas *Premise 2* is accounted through *Chapters Second* and *Third*, thus it does not need further justification. *Premise 3* is largely discussed in *Chapter Third*, whereas *Premise 4* is granted by the construction of the argument. *Premise 5* can be deduced by *Premises 1* and *4*, and some crucial aspects it introduces are grounded on Conditions 3 and 5 and the related justifications. In the end, *Premise 6* is based on *Condition 5* and the other *Premises*. All together they allow to draw the conclusion.

3.5. Closing Remark

Before closing the work, I would like to stress an important point. *This argument is not supposed to prove in any case that mental content is something different from its physical realization by the brain*. The argument only suggests that *maybe the mind-body relation is wrongly conceptualized*, and this can affect in turn the attempts of explanation of mental processing (thinking and other mental activities) and of mind-body relation like Computationalism in philosophy of mind. As I said, it seems to me that contemporary attempts of explaining mind presuppose that mind is something that comes into play when all has been done, probably because they all share the presupposition that equates describing production with explaining mental content. For if this explanatory model must work, mind must be considered a *phenomenon*, namely some deceiving appearance whose reality consists of that which realizes it, in the same way in which the sensation of heat is just the deceiving appearance of our relation with molecular motion.

In another occasion, I suggested that *phenomenal mind as such* could be *conceptualized as an inherent moment of brain activity, probably according to the ontology of states* (as opposed to ontology of effects[503]), but this implies breaking the idea that mind serves some impersonal and evolutionistic purpose: if mind contributes causally in a meaningful way to the process of which it is a step (as in the Husserlian perspective I briefly outlined), then mind is an end in itself, and must not be explained in some way different from itself, exception made for its existence, since if it is a stage of an activity, then there is a *substrate of this activity*, which cannot be mind as the totality of phenomenal and aware activity, since philosophers, especially Kant, have already argue that this is an inconsistent position. Even if the demonstration of the causal relevance of mental contents in their descriptive form leaves clueless on how phenomenal mind can concretely be causally relevant as a state given that it is a physical state, for the purpose of the autonomy of philosophy it would

[503] In cause-effect relation, causes are separated from their effect, at least conceptually, whereas in the case of states their modalities are integrated into the substrate.

be enough to demonstrate the necessity of its contribution under some respect. An argument as the one for the duplication of Concept in a suitable, generalized form may constitute a first step towards this demonstration in the context of knowledge.

References

Aizawa, K. (2019). Turing-Equivalent Computation at the "Conception" of Cognitive Science. In M. Sprevak, & M. Colombo (Eds.), *The Routledge Handbook of the Computational Mind* (pp. 65-75). Routledge: New York.

Allison, H. (2004). *Kant's Transcendental Idealism - An Interpretation and Defense.* Yale: Yale University Press.

Angell, J. (1907). The Province of Functional Psychology. *Psychological Review*, 61-91.

Aspray, W. (1985). The Scientific Conceptualization of Information. *Annals of the History of Computing, 7*(2), 117-140.

Backe, A. (2001). John Dewey and Early Chicago Functionalism. *History of Psychology, 4*(4), 323-340.

Barthes, R. (1964/1986). *Elements of Semiology.* (A. Lavers, & C. Smith, Trans.) New York: Hill and Wang.

Bechtel, W. (2008). *Mental Mechanisms: Philosophical Perspectives on Cognitive Neuroscience.* London: Routledge.

Berto, F. (2008). *Tutti Pazzi per Gödel.* Bari: Laterza.

Bickle, J. (2010). Multiple Realizability. In *Stanford Encyclopedia of Philosophy.* Retrieved from https:\\plato.stanford.edu/archives/spr2019/entries/multiple-realizability

Blanchette, P. (2018). The Frege-Hilbert Controversy. In E. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy.* doi:https://plato.stanford.edu/archives/fall2018/entries/frege-hilbert/

Boden, M. (2006). *Mind as a Machine: A History of Cognitive Science.* Oxford: Oxford University Press.

Brentano, F. (1874/1995). *Psychology from an Empirical Standpoint.* Oxford: Oxford University Press.

Brette, R., Rudolph, M., Carnevale, T., Hines, M., Beeman, D., Bower, J., . . . Djurfeld, M. (2007). Simulation of Networks of Spiking Neurons: A Review of Tools and Strategies. *Journal of Computational Neuroscience, 23*(3), 349-398.

Brysbaert, M., & Rastle, K. (2009). *Historical and Conceptual Issues in Psychology.* Harlow: Pearson Education.

Cao, R. (2019). Computational Explanations and Neural Coding. In M. Sprevak, & M.

Colombo (Eds.), *The Routledge Handbook of Computational Mind* (pp. 283-296). New York: Routledge.

Cappelletti, M., & Cipollotti, L. (2006). Unconscious Processing of Arabic Numerals in Unilateral Neglect. *Neurophysiologia, 44*(10), 1999-2006.

Cartwright, R. (1971). Identity and Substitutivity. In M. Munitz (Ed.), *Identity and Individuation* (pp. 119-133). New York: New York University Press.

Cassirer, E. (1923/1955). *Philosophy of Symbolic Forms* (Vols. 1 - Language). (R. Manheim, Trans.) London: Yale University Press.

Cassirer, E. (1923/1955). *The Philosophy of Symbolic Forms* (Vols. Vol.1 – Language). (R. Manheim, Trans.) Yale, Yale University Press.

Cassirer, E. (1929/1957). *Philosophy of Symbolic Forms* (Vols. 3 - The Phenomenology of Knowledge). (R. Manheim, Trans.) London: Yale University Press.

Church, A. (1936). An Unsolvable Problem of Elementary Number Theory. *American Journal of Mathematics, 58*(2), 345-363.

Churchland, P. M. (1992). *A Neurocomputational Perspective: The Nature of Mind and the Structure of Science.* Cambridge (Massachusetts): Cambridge University Press.

Churchland, P. S., Koch, C., & Sejnowski, T. J. (1990). What Is Computational Neuroscience? In E. Schwarz (Ed.), *Computational Neuroscience* (pp. 46-55). Cambridge (Massachusetts): Cambridge University Press.

Copeland, J. (2004). The Church-Turing Thesis. *Neuroquantology, 2*, 101-115.

Costall, A. (2004). From Darwin to Watson (and Cognitivism) and Back Again: The Principle of Animal-Environment Mutuality. *Philosophy and Behavior, 32*, 179-195.

Cowan, J. (1990). Discussion: McCulloch and Pitts and Related Neural Nets from 1943 to 1989. *Bullettin of Mathematical Biology, 52*(1-2), 73-97.

Crane, T. (1990). "The Language of Thought": No Syntax without Semantics. *Mind and Language, 5*(3), 187-212.

Crane, T. (2016). *The Mechanical Mind: A Philosophical Introduction to Minds, Machines and Mental Representations.* New York: Routledge.

Descartes, R. (1619/1998). *Regualae ad Directionem Ingenii.* (G. Heffernan, Trans.) Amsterdam: Rodopi.

Dias, F., Antunes, A., & Mota, A. (2004). Artificial Neural Networks: A Review of

Commercial Hardware. *Engineering Applications of Artificial Intelligence, 17*(8), 945-952.

Dimitrov, A., Lazar, A., & Victor, J. (2011). Information Theory in Neuroscience. *Journal of Computational Neuroscience, 30*, 1-5.

Dretske, F. I. (1981/1982). *Knowledge and the Flow of Information.* Cambridge (Massachusetts): MIT Press.

Dreyfus, H. (1968). Cybernetics as the Last Stage of Metaphysics. *Akten des XIV Internationalen Kongresses für Philosophie, 2*, pp. 493-499.

Egan, F. (2019). The Nature and Function of Content In Computational Models. In M. Sprevak, & M. Colombo (Eds.), *The Routledge Handbook of the Computational Mind* (pp. 247-258). New York: Routledge.

Eigenmann, R., & Lilja, D. (1999). Von Neumann Computers. In J. Webster (Ed.), *Wiley Enciclopedia of Electrical and Electronical Engineering* (Vol. 23, pp. 387-400).

Elgin, C. (2010). Telling Instances. In R. H. Frigg (Ed.), *Beyond Mimesis and Convention. Representation in Art and Science* (pp. 1-17). Berlin: Springer.

Fantini, B. (2000). La Neurofisiologia e Neurobiologia nel Primo Novecento. In E. Bellone, U. Bottazzini, B. Fantini, A. La Vergata, S. Poggi, & E. Torracca (Eds.), *Storia della Scienza Moderna e Contemporanea* (Vol. 5, pp. 479-491). Milano (Italy): TEA Edizioni.

Fitch, A. (1944). Review of a Logical Calculus of the Ideas Immanent in Nervous Activity by Warren S. McCulloch and Walter Pitts. *Journal of Symbolic Logic*(9), 49-50.

Floridi, L. (2004). Information. In L. Floridi (Ed.), *The Blackwell Guide to Philosophy of Computing and Information* (pp. 40-61). Oxford (UK): Blackwell Publishing.

Floridi, L. (2008). The Methods of Level of Abstraction. *Minds and Machines, 18*(3), 303-329.

Fodor, J. (1975). *The Language of Thought.* New York: Thomas Y. Cromwell Company.

Fodor, J. (1998). *Concepts - Where Cognitive Science Went Wrong.* Oxford: Clarendon Press.

Fodor, J. A., & Pylyshyn, Z. (1988). Connectionism and Cognitive Architecture: A Critical Analysis. *Cognition, 28*(1-2), 3-71.

Frege, G. (1878/1967). Begriffsschrift, a Formalized Language of Pure Though Modelled Upon the Language of Arithmetic. In J. Van Heijenoort (Ed.), *From Frege to Godel*

- *A Source Book in Mathematical Logic, 1879-1931* (S. Bauer-Mengelberg, Trans., pp. 1-82). Cambridge (Massachusetts): MIT Press.

Frege, G. (1884/1960). *The Foundations of Arithmetic - A Logico-Mathematical Enquiry Into the Concept of Number*. New York: Harper Books.

Frege, G. (1885/1991). On Formal Theories of Arithmetics. In B. McGuinness (Ed.), *Collected Papers on Mathematics, Logic and Philosophy* (pp. 112-121). Glasgow: Blackwell Publishing.

Frege, G. (1892/1960). On Sense and Reference. In P. Geach, & M. Black (Eds.), *Translations from the Philosophical Writings of Gottlob Frege* (M. Black, Trans., pp. 56-78). Oxford: Basil Blackwell.

Frege, G. (1903/1984). On the Foundations of Geometry (First Series). In B. McGuinness (Ed.), *Collected Papers in Mathematics, Logic, and Philosophy* (pp. 273-284). Oxford: Blackwell Publishing.

Gallagher, S., & Zahavi, D. (2008). *The Phenomenological Mind - An Introduction to Cognitive Science*. New York: Routledge.

Gardner, H. (1985). *The Mind New Science - A Histhory of the Cognitive Revolution.* New York: Basic Books.

Gershman, S., Horvitz, E., & Tenenbaum, J. (2015). Computational Rationality: A Converging Paradigm for Intelligence in Brains, Minds and Machines. *Science, 349*(6245), 273-278.

Graupe, D. (2007). *Principles of Artificial Neural Network.* Singapore: World Scientific Publishing.

Green, C. (2009). Darwinian Theory, Functionalism, and the First American Psychological Revolution. *American Psychologist, 64*(2), 75-83.

Grollier, J., Querlioz, D., & Stiles, M. (2016). Spintronic Nanodevices for Bioinspired Computing. *Proceedings of the IEEE, 104*(10), 2024-2039. doi:10.1109/JPROC.2016.2597152

Harnad, S. (1990). The Symbol Grounding Problem. *Physica D: Nonlinear Phenomena, 42*(1-3), 335-346.

Hartley, R. (1928). Transmission of Information. *Bell Labs Technical Journal, 7*(3), 535-563.

Haugeland, J. (1989). *Artifical Intelligence: The Very Idea.* Cambridge (Massachusetts):

Bradford Book.

Haykin, S. (2009). *Neural Networks and Learning Machines.* New York: Pearson.

Hilbert, D. (1918/2007). Axiomatic Thought. In W. Ewald (Ed.), *From Kant to Hilbert - A Source Book in the Foundations of Mathematics* (Vol. 2, pp. 1107-1115). Oxford: Clarendon Press.

Hilbert, D. (1922/2007). The New Grounding of Mathematics. First Report. In W. Ewald (Ed.), *From Kant to Hilbert - A Source Book in the Foundations of Mathematics* (Vol. 2, pp. 1115-1132). Oxford: Clarendon Press.

Hilbert, D. (1925/1967). On the Infinite. In J. van Heijenoort (Ed.), *From Frege to Godel - A Source Book in Mathematical Logic, 1879-1931* (pp. 367-392). Cambridge (Massachusetts): Harvard University Press.

Hinton, G., McClelland, J., & Rumelhart, D. (1987). Distributed Representations. In J. McClelland, & D. Rumelhart (Eds.), *Parallel Distributed Processing. Explorations in the Microstructures of Cognition* (Vols. Vol.1 - Foundations, pp. 77-109).

Husserl, E. (1920/2001). *Logical Investigations.* (J. Findlay, Trans.) New York: Routledge.

Jackson, F. (1986). What Mary Didn't Know. *The Journal of Philosophy, 83*(5), 291-295.

Jain, A., Mao, J., & Mohiuddin, K. (1996). Artificial Neural Networks: A Tutorial. *Computer, 29*(3), 31-44.

James, W. (1890/2007). *Principles of Psychology*. London: Harvard University Press.

Jeong, H. &. (2018). Memristor Devices for Neural Networks. *Physica D: Applied Physics, 52*(2). Retrieved from https://iopscience.iop.org/article/10.1088/1361-6463/aae223/pdf

Johnson-Laird, P., & Wason, P. C. (Eds.). (1977). *Thinking: Reading in Cognitive Science.* Cambridge: Cambridge University Press.

Kandel, E. R., Schwartz, J. H., Jessel, T. M., Siegelbaum, S. A., & Hudspeth, A. J. (Eds.). (2013). *Principles of Neuroscience.* New York: McGraw-Hill.

Kant, I. (1787/1998). *Critique of Pure Reason.* (P. A. Guyer, Ed.) Cambridge: Cambridge University Press.

Kaplan, D. (1973). Bob and Carol and Ted and Alice. In J. S. Hintikka (Ed.), *Approaches to Natural Language* (pp. 490-518). Dordrecht: Springer.

Kaufmann, F. (1949). Cassirer's Theory of Scientific Knowledge. In P. Schlipp (Ed.), *The Philosophy of Ernst Cassirer* (pp. 185-214). Evanston (Illinois): Library of Living

Philosophers Inc.

Kim, J. (1998). *Mind in a Physical Word.* Cambridge: MIT Press.

Kim, J. (2003). The American Origin of Philosophical Naturalism. *Journal of Philosophical Research, 28*(Supplement), 83-98.

Kim, J. (2005). *Physicalism, or Something Near Enough.* Princeton: Princeton University Press.

Kim, J. (2010). *Philosophy of Mind.* Boulder (Colorado): Westview Press.

Kleene, S. (1951). *Representation of Events in Nerve Nets and Finite Automata*. Retrieved from www.rand.org/content/dam/rand/pubs/research-memoranda/2008/RM704.pdf.

Lashley, K. S. (1923a). The Behavioristic Interpretation of Consciousness I. *Psychological Review, 3*(4), 237-272.

Lashley, K. S. (1923b). The Behavioristic Interpretation of Consciousness II. *Psychological Review, 3*(5), 329-353.

Lashley, K. S. (1951). The Problem of Serial Order in Behavior. In L. Jeffres (Ed.), *Cerebral Mechanisms in Behavior: the Hixon Symposium* (pp. 112-146). New York: Hafner Publishing.

Leeuw, K. d., Moore, E., Shannon, C., & Shapiro, N. (1956/1993). Computability by Probabilistic Machines. In N. Sloane, & A. Wyner (Eds.), *Claude Shannon - Collected Papers* (pp. 742-771). Piscataway (New Jersey): John Wiley and Sons Publication.

Lovelace, A. (1843). Notes by the Translator. In R. Taylor (Ed.), *Scientific, Memoirs, Selected From the Transactions of Foreign Academies of Science and Learned Societies and Foreign Journals* (pp. 691-731). London: Richard and John Taylor.

MacCormick, J. (2018). *What Can Be Computed? A Practical Guide to the Theory of Computation.* Princeton: Princeton University Press.

Marr, D. (1979/1982). *Vision - A Computational Investigation into the Human Representation and Processing of Visual Information.* New York: W.H. Freeman and Company.

McCulloch, W., & Pitts, W. (1943). A Logical Calculus of Ideas Immanent to Nervous Activity. *Bullettin of Mathematical Biophysics, 5*, 115-132.

Milkowski, M. (2014). *Explaining the Computational Mind.* Cambridge (Massachusetts):

Cambrige University Press.

Miller, D. (1947). Explanation Versus Description. *The Philosophical Review, 56*(3), 306-312.

Minsky, M. (1967). *Finite and Infinite Machines.* Englewood Cliffs (USA, NJ): Prentice-Hall Inc.

Moore, J. (2003). Explanation and Description in Traditional Neobehaviorism, Cognitive Psychology, and Behavior Analysis. In L. K.A., & C. P.N. (Eds.), *Behavior Theory and Philosophy* (pp. 13-39). Boston: Springer.

Nagel, E. (1974). What Is Like to Be a Bat? *The Philosophical Review, 83*(4), 435-450.

Nannini, S. (2007). *Naturalismo Cognitivo - Per una Teoria Materialistica della Mente.* Macerata: Quodlibet.

Nannini, S., & Sandkühler, H. J. (Eds.). (2000). *Naturalism in Cognitive Sciences and the Philosophy of Mind.* Bern: Peter Lang Publishing.

Newell, A. (1980). Physical Symbol Systems. *Cognitive Science, 4*, 135-183.

Nicasi, S. (2000). La Psicologia dal 1930 al 1950. In E. Bellone, U. Bottazzini, B. Fantini, A. La Vergata, S. Poggi, & E. Torracca (Eds.), *Storia della Scienza Moderna e Contemporanea* (Vol. 6, pp. 1009-1030). Milano: TEA Edizioni.

Nyquist, H. (1924). Certain Factors Affecting Telegraph Speed. *Transactions of the American Institute of Electrical Engineers, 47*(2), 324-346.

O'Reilly, R. C., & Munakata, Y. (2000). *Computational Explanations in Cogntive Neuroscience: Understanding the Mind By Simulationg the Brain.* Cambridge (Massachusetts): Cambridge University Press.

Patterson, D., & Hennessy, J. (2013). *Computer Organization and Design: The Hardware/Software Interface.* Oxford: Morgan Kaufmann (Elsevier).

Patton, L. (2014). Hilbert's Objectivity. *Historia Mathematica, 41*, 188-203.

Perlman, M. D. (2016). The Modern Philosophical Resurrection of Teleology. *The Monist, 87*(1), 3-51.

Piccinini, G. (2004). The First Computational Theory of Mind and Brain: A Close Look to McCulloch and Pitts's "Logical Calculus of Ideas Immanent in Nervous Activity". *Synthese, 141*(2), 175-215.

Piccinini, G. (2006). Computation without Representation. *Philosophical Studies, 137*(2), 205-241.

Piccinini, G. (2008). Some Neural Nets Compute, Others Don't. *Neural Networks, 21*, 311-321.

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

Piccinini, G. (2018). Computation and Representation in Cognitive Neuroscience. *Minds and Machines, 28*, 1-6.

Piccinini, G., & Scarantino, A. (2010). Computation vs Information Processing. *Studies in History and Philosophy of Science, 41*(3), 237-246.

Piccinini, G., & Scarantino, A. (2010). Computing vs Information Processing: Why Their Difference Matters to Cognitive Science. *Studies in Historiy and Philosophy of Science, 41*(3), 237-246.

Pitt, D. (2004). The Phenomenology of Cognition, or "What Is Like to Think That P?". *Philosophy and the Phenomenological Research, 69*(1), 1-36.

Poggi, S. (2000). Lo Sviluppo della Psicologia dal 1900 al 1930. In E. Bellone, U. Bottazzini, B. Fantini, A. La Vergata, S. Poggi, & E. Torracca (Eds.), *Storia della Scienza Moderna e Contemporanea* (Vol. 5, pp. 511-533). Milano (Italy): TEA Edizioni.

Poggi, S., & Nicasi, S. (2000). La Psicologia e il Dibattito Psichiatrico. In E. Bellone, U. Bottazzini, B. Fantini, A. La Vergata, S. Poggi, & E. Torracca (Eds.), *Storia della Scienza Moderna e Contemporanea* (Vol. 5, pp. 511-533). Milano (italy): TEA Edizioni.

Priestley, M. (2011). *A Science of Operations. Machines, Logic and the Invention of Computing.* New York: Springer.

Ramsey, W. (2007). *Representation Revised.* Oxford: Oxford University Press.

Resnik, M. (1974). The Frege-Hilbert Controversy. *Philosophy and Phenomenological Research, 34*(3), 386-403.

Robič, B. (2015). *The Foundations of Computability Theory.* Berlin: Spinger.

Rogers, T. M. (2014). Parallel Distributed Processing at 25: Further Explorations in the Microstructure of Cognition. *Cognitive Science, 38*(6), 1024-1077.

Rojas, R. (1996). *Neural Networks - A Systematic Introduction.* Berlin: Springer.

Rosenblueth, A., Wiener, N., & Bigelow, J. (1943). Behavior, Purpose and Teleology. *Philosophy of Science, 10*(1), 18-24.

Samuels, R. (2019). Classic Computational Models. In M. Sprevak, & M. Colombo (Eds.), *The Routledge Handbook of the Computational Mind* (pp. 103-119). New York: Routledge.

Schlatter, M., & Aizawa, K. (2008). Walter Pitts and "A Logical Calculus". *Synthese, 162*(2), 235-250.

Searle, J. (1980). Minds, Brains and Programs. *Behavioral and Brain Sciences, 3*(3), 417-434.

Shagrir, O. (2018). In Defense of the Semantic View of Computation. *Synthese*, 1-26. doi:https://doi.org/10.1007/s11229-018-01921-z

Shannon, C. E. (1949/1960). The Mathematical Theory of Communication. In C. Shannon, & W. Weaver (Eds.), *The Mathematical Theory of Communication* (pp. 31-125). Chicago: University of Illinois Press.

Shannon, C. E. (1950/1993). Communication Theory - Exposition of Fundamentals. In N. Sloane, & A. Wyner (Eds.), *Claude Elwood Shannon - Collected Papers* (pp. 173-176). Piscataway (New Jersey): John Wiley and Sons Publication.

Shannon, C. E. (1968/1993). Information Theory. In N. Sloane, & A. Wyner (Eds.), *Claude Elwood Shannon - Collected Papers* (pp. 212-220). Piscataway (New Jersey): John Wiley & Sons Publication.

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

Sieg, W. (1998/2016). Calculation by Man and Machine - Conceptual Analysis. In W. Sieg, R. Sommer, & C. Talcott (Eds.), *Reflections on the Foundation of Mathematics - Essays in Honor of Solomon Feferman* (pp. 390-409). Cambridge: Cambridge University Press.

Sieg, W. (2009). Hilbert's Proof Theory. In D. Gabbay, & J. Woods (Eds.), *Handbook of the History of Logic - Logic from Russell to Church* (Vol. 5, pp. 321-384). Amsterdam: Elsevier.

Sieg, W. (2018). What Is the Concept of Computation? In F. Manea, R. Miller, & D. Nowotka (Eds.), *Sailing Routes in the World of Computation. CiE 2018* (pp. 386-396). Berlin: Springer.

Siegelmann, H., & Sontag, E. (1991). Turing Computability with Neural Nets. *Applied Mathematics Letters, 4*(6), 77-80.

Siegelmann, H., & Sontag, E. (1995). On the Computational Power of Neural Nets. *Journal*

of Computer and System Sciences, 50, 132-150.

Silc, J., Robič, B., & Ungerer, T. (1999). *Processor Architecture - From Dataflow to Superscalar and Beyond.* New York: Springer.

Simon, H., & Newell, A. (1971). Human problem solving: The state of the theory in 1970. *American Psychologist, 26*(2), 145–159. doi:https://doi.org/10.1037/h0030806

Skinner, B. F. (1953/2005). *Science and Human Behavior* (Burrhus Friederic Foundation ed.). Boston: Pearson Education. Retrieved from http://www.behaviorpedia.com/wp-content/uploads/2013/01/Science_and_Human_Behavior.pdf

Skinner, B. F. (1974/1976). *About Behaviorism.* New York: Vintage Books.

Skinner, B. F. (1977). Why I Am Not a Cognitive Psychologist. *Behaviorism, 5*(2), 1-10.

Skinner, B. F. (1990). Can Psychology Be a Science of Mind? *American Psychologist, 45*(11), 1206-1210.

Skrzypek, J. (Ed.). (1993). *Neural Network Simulation Environments.* New York: Springer.

Sloman, A. (2002). The Irrelevance of Turing Machines to AI. In M. Scheutz (Ed.), *Computationalism - New Directions* (pp. 87-127). Cambridge (Massachusetts): Bradford Book.

Sloman, A. (2018). Huge, but Unnoticed, Gaps Between AI and Natural Intelligence. In V. Müller (Ed.), *Philosophy and Theory of Artificial Intelligence 2017* (pp. 92-105). Cham (Switzerland): Springer.

Sluga, H. D. (1980/1999). *Gottlob Frege.* London: Routledge.

Soare, R. I. (1996). Computability and Recursion. *The Bullettin of Symbolic Logic, 2*(3), 284-321.

Solmitz, W. (1949). Cassirer On Galileo: an Example of Cassirer's Way of Thought. In P. Schlipp (Ed.), *The Philosophy of Ernst Cassirer* (pp. 731-765). Evanston (Illinois): Library of Living Philosophers Inc.

Sprevak, M. (2011). William M. Ramsey Representation Reconsidered. *The British Journal for the Philosophy of Science, 62*(3), 669-675.

Sprevak, M. (2019). Triviality Arguments About Computational Implementation. In M. Sprevak, & M. Colombo (Eds.), *The Routledge Handbook of the Computational Mind* (pp. 175-191). New York: Routledge.

Strawson, G. (1994). *Mental Reality.* Cambridge: Cambridge University Press.

Tolman, E. (1925). Purpose and Cognition: The Determiners of Animal Learning. *Psychological Review, 32*(4), 285-297.

Tolman, E., & Honzik, C. (1930). Introduction and Removal of Reward and Maze Performance in Rats. *University of Chicago Publications in Psychology, 4*(17), 257-275.

Tonneau, F. (2011). Metaphor and Truth: A Review of Representation Reconsidered By W.M. Ramsey. *Philosophy and Behavior, 39-40*(2011-2012), 331-343.

Turing, A. M. (1936). On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society, 2*(1), 230-265.

Turing, A. M. (1950/2004). Computing Machinery and Intelligence. In B. Copeland (Ed.), *The Essential Turing* (pp. 432-464). Oxford: Oxford University Press.

Turner, R., Angius, N., & Primiero, G. (2019). The Philosophy of Computer Science. In E. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy (Spring 2019 Edition).* Retrieved from https://plato.stanford.edu/archives/spr2019/entries/computer-science/

Von Neumann, J. (1951). The General and Logical Theory of Automaton. In L. Jeffres (Ed.), *Cerebral Mechanisms in Behavior: the Hixon Symposium* (pp. 1-41). New York: Hafner Publishing.

Watson, J. (1913). Psychology as the Behaviorist Views It. *Psychological Review, 20*, 158-177.

Watson, J. (1914/1931). *Behaviorism.* London: Keagan Paul, Trench, Traubner and Co.

Weaver, W. (1949/1960). Introductory Note on the General Setting of the Analytical Communication Studies. In *The Mathematical Theory of Communication* (pp. 3-28). Chicago: University of Illinois Press.

Wiener, N. (1948). Cybernetics. *Scientific American, 179*(5), 14-19.

Wiener, N. (1961/1985). *Cybernetics, or Control and Communication in The Animal and The Machine.* Cambridge (Massachusetts): MIT Press.

Wittgentstein, L. (1953/2009). *Philosophical Investigations.* (P. Hacker, & J. Schulte, Trans.) Chilcester (UK): Blackwell Publishing.

Yang, J., Strukov, D., & Stewart, D. (2012). Memristive Devices for Computing. *Nature Nanotechnology, 8*(1), 13-24.

Zhu, J., Milne, G., & Gunther, B. (1999). Towards an FPGA Based Reconfigurable Computing Environment for Neural Network Implementations. *IET Conference Proceedings, 1999*, (pp. 661-666). doi:10.1049/cp:19991186