

Stanisław Ambroszkiewicz

ORCID: 0000-0002-8478-6703

University of Siedlce
Faculty of Exact and Natural Sciences
Institute of Computer Science
ul. 3 Maja 54, 08-110 Siedlce, Poland

stanislaw.ambroszkiewicz@uws.edu.pl

The architecture of human mind, and non-von Neumann computer architecture

DOI: 10.34739/si.2025.32.01

Abstract. Structural plasticity in the brain (i.e. rewiring the connectome) may be viewed as mechanisms for dynamic reconfiguration of neural circuits. First order computations in the brain are done by static neural circuits, whereas higher order computations are done by dynamic reconfigurations of the links (synapses) between the neural circuits. Static neural circuits correspond to first order computable functions. Synapse creation (activation) between them correspond to the mathematical notion of function composition. Functionals are higher order functions that take functions as their arguments. The construction of functionals is based on dynamic reconfigurations of function compositions. Perhaps the functionals correspond to rewiring mechanisms of the connectome. The architecture of human mind is different than the von Neumann computer architecture. Higher order computations in the human brain (based on functionals) may suggest a non-von Neumann computer architecture, a challenge posed by John Backus in 1977 [7]. The presented work is a substantial extension and revision of [2].

Keywords: higher order computations, rewiring the connectome, non-von Neumann computer architecture

1 Introduction

Leopold et al. (2019) [21]: “One of the great challenges in the study of the brain is to synthesize a large number of details into principles for understanding. While many details are known about the cerebral cortex, our level of understanding about its overarching architectural and functional principles remains, arguably, primitive.”

Gedankenexperiment: a backward time travel of a computer: A contemporary computer was moved into the XIX-th century so that scientists could make experimental research. Actually, the idea underlining the functioning of a computer is extremely simple; it is the von Neumann computer architecture. Would it be possible for the scientists of nineteenth century to discover the idea by examining the electric circuits and their complex functioning of the working computer system consisting of: monitor, a motherboard, a CPU, a RAM, graphic cards, expansion cards, a power supply, an optical disc drive, a hard disk drive, a keyboard and a mouse? What about BIOS and operating system as well as many applications installed?

Perhaps the Gedankenexperiment may serve as a metaphor of the research on the human brain functioning. Although great discoveries and achievements have been made in Neurobiology, the basic mechanisms (idea) underling the human brain functioning are still a great mystery.

The research on computational models of neural circuits is well established starting with McCulloch-Pitts networks [25] via the Hopfield model ([15] and [16]) to recurrent neural networks (RNNs). It seems that RNNs adequately represent the computations done in the human brain by the real neuron networks. From the Computer Science point of view, RNNs are Turing complete (Siegelmann and Sontag [33]), i.e., every computable function may be represented as a RNN. However, Turing machine is a flat model of computation. There are also higher order computations, i.e. computable functionals where arguments (input) as well as values (output) are functions.

The Virtual Brain (TVB [32], www.thevirtualbrain.org) project aims at building a large-scale simulation model of the human brain. It is supposed that brain function may emerge from the interaction of large numbers of neurons, so that, the research on TVB may contribute essentially to our understanding of the spatiotemporal dynamics of the brain’s electrical activity. However, after more than 10 years of intensive and expensive research and the expenditure of hundreds of millions of euros, it is still unclear how this activity may contribute to the comprehension of the principles of the human mind functioning.

Adolphs 2015 [1]: “Some argue that we can only understand the brain once we know how it could be built. Both evolution and development describe temporally sequenced processes whose final expression looks very complex indeed, but the underlying generative rules may be relatively simple ... ”

Another interesting approach is due to Juergen Schmidhuber: “The human brain is a recurrent neural network (RNN): a network of neurons with feedback connections”; see <http://people.idsia.ch/~juergen/rnn.html> . Indeed, real neural circuits can be modeled as (continuous time) RNNs. Despite the enormous complexity of a hypothetical RNN modeling the human brain, there is a paradox here because (continuous time) RNNs are nonlinear dynamic systems. It means that RNNs are high level mathematical abstractions (of human mind) involving the notion of space-time Continuum that comprises actual infinity. These very abstractions

are created in the human brain (consisting of a finite number of cells), i.e. the notions related to space-time continuum are represented (in the brain) as finite structures.

Some parts of the connectome may and should be considered as modules responsible for particular (elementary) cognitive functions of the brain. This very modularity reduces considerably the complexity. Once the modules are distinguished as functions with clearly defined input and output, it gives rise to compose them. The composition is, in turn, the basic mechanism for constructing higher order functionals. However, it seems that RNNs still lack the modularity and ability to compose the modules. Perhaps, if the notions of modularity and computable functionals were introduced to RNNs, they could model the higher order computations as dynamic formation and reconfigurations of the links (synapses) between the neurons.

Recent advances in Neurobiology prove that reconfigurations of functional brain networks are responsible for higher organization of information processing in the neuronal connectome. However, generic mechanisms for dynamic reconfigurations of connections between neural circuits are not known. Since information processing in the brain is identified with computations performed by neural circuits, higher organization of information processing in the brain corresponds to higher order computations in Computer Science. They are based on the notion of functionals that take functions as arguments and return functions as values. It is well known that constructions of such functionals are based on dynamic creations and reconfigurations of data-flow connections between elementary functions. This fact strongly relates functionals to the generic mechanisms for dynamic reconfigurations of connections between neural circuits in the brain.

The foundations of the mind functioning might be ingenious in its simplicity although the underlying biological mechanisms are extremely complex and sophisticated. Hence, in order to model neural circuits and the mechanisms responsible for structural changes in the neuronal connectome, let us use much more simple (than RNN) primitive notions from Mathematics and Computer Science, i.e. the computable functions and computable functionals. Since Mathematics is a creation of the human mind, the Foundations of Mathematics may shed some light on the principles of the brain functioning. That is, the basic mathematical notions can be recognized as concrete mental structures, and then the corresponding mechanisms of the human brain can be discovered.

Evidently, the architecture of human mind is different than von Neumann computer architecture. Perhaps a non-von Neuman architecture (postulated by John Backus [7]) may result in implementation of the functionals in hardware in the very similar way as it is done in the human brain.

The fundamental assumption of the presented work relates functionals (as higher order computing) to higher order information processing in the human brain. This may give rise to expect that higher order computable functions (functionals) have counterparts in the human brain.

Composition (as link creation) is the basic operation for function constructions as well as for construction of higher order functions (functionals). This very composition corresponds to synapse creation (activation) in the brain.

Functionals are construction (as mental structures) of the pure intellect of the human mind.

Hence, the following hypothesis seems to be reasonable:

The mechanisms that are responsible for the higher order organization of information processing in the human brain correspond to computable functionals.

There are experimental evidences confirming the hypothesis, e.g. Bertolero (2015) et al. [8], Braun et al. (2015) [9], and Park and Friston (2013) [31].

One of the main goals of this article is to at least substantiate this hypothesis by sketching the calculus of functionals introduced in [3].

The principles for constructing such mechanisms (functionals) are extremely simple like the ones that are basis for the von Neumann computer architecture. Contemporary computers (built on the the von Neumann computer architecture von Neumann 1958 [27] and 1966 [28]) are sophisticated in their functionality. In the same spirit, human brain and its functioning (self-consciousness, cognitive structures, behavior, learning, emotions, feelings, relation with other humans, etc.) are sophisticated despite the fact that they are based on extremely simple principles.

Since the architecture of human brain is definitely different than von Neumann computer architecture, the mechanisms for rewiring the connectome (i.e. the meta-plasticity) may give rise to develop a non-von Neumann computer architecture and a corresponding function-level programming language postulated by John Backus 1977 [7].

Perhaps a non-von Neuman architecture (postulated by John Bacus [7]) may result in implementation of the functionals in hardware in the very similar way as it is done in the human brain. We will explore this idea in Section 4. This is the second goal of this article.

2 Neural circuits, computable functions and functionals

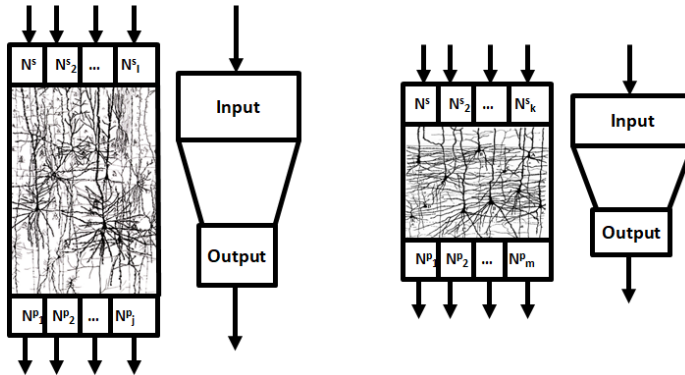


Figure 1. Abstraction: neural circuits as functions. A function consists of input (socket), and output (plug)

Before going into details, several assumptions are to be made. The first one is that elementary neural circuits (corresponding to functional units of the brain) can be distinguished. The second assumption is that any such circuits (at least temporary) has clearly identified input (dendrite spines of some postsynaptic neurons) and output (axons of some presynaptic

neurons). It means that the output is exactly determined by the input. The third assumption is that such circuits can be composed by a linking the output of one circuit to the input of another circuit; it may be done by (activating) creating a (silent) synapse connecting an axon (of the output of one circuits) to a dendrite spine of the input of the other circuit. If the above assumptions can be verified experimentally, then the following considerations make sense. However, from the conceptual point of view, they may also be of some interest to Neurobiology.

If the above assumption are taken as granted, then a neural circuit can be represented as a first order function defined on natural numbers, see Fig. 1. That is, spike sequences (bursts), generated by a neuron, may be interpreted as natural numbers in the unary code, input of the circuit as arguments whereas output as values of the function. Note, that this is a static (one shot) representation of neural circuits. It means that one output is produced form one input.

However, if a circuit is to be considered in a time extent so that for consecutive inputs, it produces a sequence of outputs, then dynamic behavior of the circuit may be represented either as a RNN or as a sequence of interrelated copies of the function representing the circuit; for details see [3].

Simple operations on functions may have their counterparts as operations on neural circuits. Given two functions f and g (from natural numbers into natural numbers), the new function h defined as $h(x) = g(f(x))$ may serve as an example, see Fig. 2. If f^c and g^c denote corresponding neural circuits, then the circuit corresponding to function h may be created by establishing (activating) synapses between input neurons of g^c , and the output neurons of f^c . This may correspond roughly to the synaptic meta-plasticity. It is interesting (however, not surprising) that this very synapse creation corresponds to a basic notion of Mathematics, i.e. to the function composition.

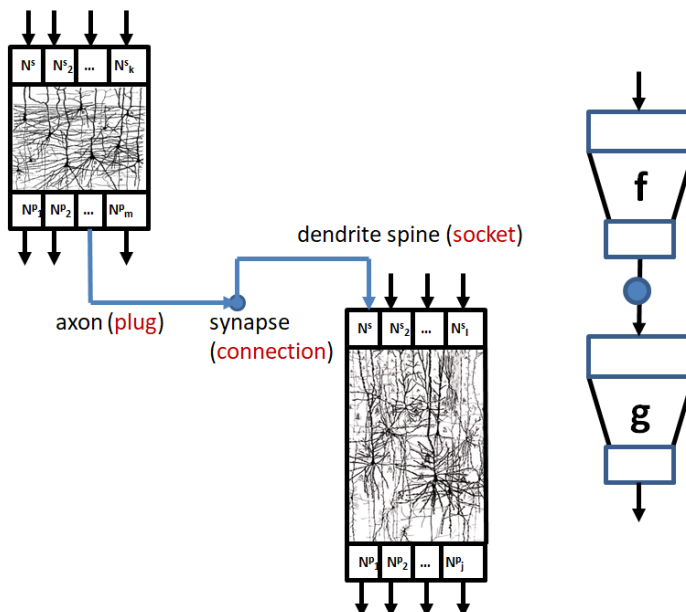


Figure 2. Connection as composition of two functions

Sockets and **plugs** are the crucial notions. A function consists of input, body and output, see Fig. 1. Input may consist of multiple sockets, whereas output may consist of multiple plugs. A plug-socket directed link may correspond to synapse as connection of axon and dendrite.

There are also higher order functions (called functionals) where arguments as well as values may be functions. It is also not surprising that these higher level functionals can be constructed by establishing links between plugs and sockets.

Each function is of some type. Since the natural numbers (finite sequences (bursts) of spikes) are assumed as the basic type (denoted by N), a type of first order functions is of the form

$$(N^{s_1}; N^{s_2}; \dots; N^{s_k}) \rightarrow (N^{p_1}; N^{p_2}; \dots; N^{p_m})$$

where $(N^{s_1}; N^{s_2}; \dots; N^{s_k})$ denotes different sockets of the input, whereas $(N^{p_1}; N^{p_2}; \dots; N^{p_m})$ denotes different plugs of the output. This type may be realized as a board consisting of sockets and plugs.

It seems that second (and higher) order computations in the brain are done by dynamic (re)configurations of links (synapses) between the neural circuits. Although the links are established between concrete neurons, these neurons belong to fixed circuits, so that (from functional point of view) the links are between circuits and correspond to circuit compositions.

Let us take as granted that higher order computation in the human brain is based on dynamic creating, composing, and reconfiguring neural circuits. At the bottom level it is realized by creating new synapses or activate existing silent synapses (see e.g. Kanold et al. 2019 [19]); this corresponds to function composition. Since the function composition is the basis for construction of the higher order functions (functionals), the mechanisms responsible for dynamic synapse creation and activation correspond (can be abstracted) to functionals.

Actually, function composition can be abstracted to a second order function (functional) that takes (as input) two first order functions, a plug of one function and a socket of the second function; then it returns (as the output) a first order function as a composition of these two functions.

We may wonder how such generic composition is realized in the brain. First of all, the circuits to be composed must be discriminated, and then passed, as parameters, to the composition mechanism.

Functionals (by their construction) have hierarchical structure that might correspond to the hierarchical organization of cortical and thalamic connectivity, see Harris et al, (2019) [14]. Hence, the following hypothesis seems to be justified.

The primitive rules for construction of the computable functionals may have their counterparts in the human brain.

2.1 A sketch of formal framework for constructing higher order computation based on functionals

Turing machines and partial recursive functions are not concrete constructions. Their definitions involve actual infinity, i.e. infinite type for Turing machines, and minimization operator μ for partial recursive functions. This results in possibility of non terminating computations that are abstract notions and have no grounding in the human brain.

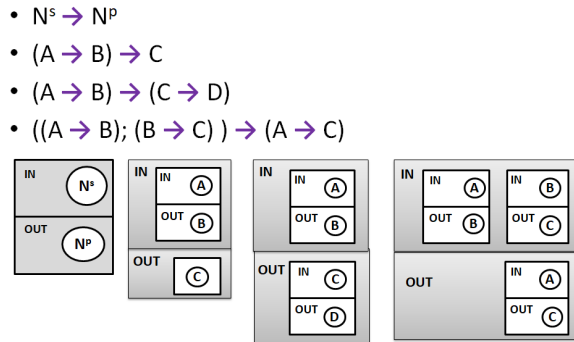


Figure 3. Function type as board of sockets and plug. Higher order types as nested boards of sockets and plugs

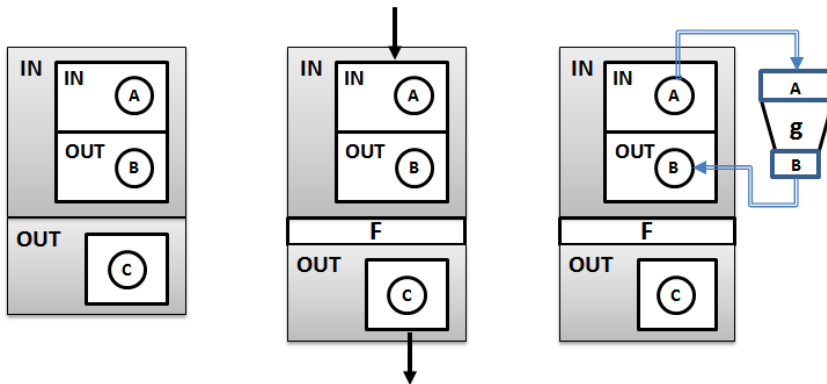


Figure 4. Higher order application of functional F of type $(A \rightarrow B) \rightarrow C$ to a function $g : A \rightarrow B$. The result $F(g)$ is an object of type C

The proposed approach is fully constructive, and if restricted only to first order computable functions, it correspond to the general recursive function according to the Herbrand-Gödel definition.

At the basic level it consists of some primitive types, primitive functions and type constructors, i.e. the type of natural numbers, the successor function, constant functions, projections, constructors for product and function type. The key primitive functionals correspond to application, composition, copy and iteration. It is crucial that these functionals can be constructed by (dynamic, in the case of iteration) establishing links between plugs (corresponding to output types) and sockets (corresponding to input types).

At the higher level of the approach, types are considered as objects, i.e. constructed as **boards of plugs and sockets**. This gives rise to introduce relations (according to the propositions-as-types correspondence of Curry-Howard), and polymorphism.

Hence, it is important to grasp the constructions of the boards as higher order types. The type of functions from natural numbers into natural numbers (denoted by $N^S \rightarrow N^P$) may be realized as a simple board consisting of a socket and a plug, see Fig. 3 where also types of higher order are presented. Note that for the type $(A \rightarrow B) \rightarrow C$, the input $A \rightarrow B$ becomes the socket. For the type $(A \rightarrow B) \rightarrow (C \rightarrow D)$, the output $C \rightarrow D$ becomes the plug.

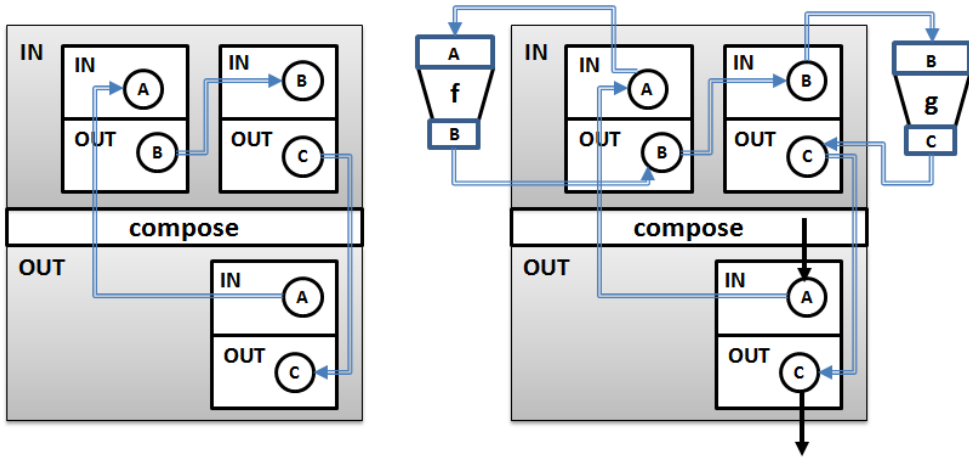


Figure 5. The functional $Comp$ of type $((A \rightarrow B); (B \rightarrow C)) \rightarrow (A \rightarrow C)$. Input objects are: f of type $A \rightarrow B$, and g of type $B \rightarrow C$. When applied to $Comp$, the output object is a function of type $A \rightarrow C$

Application of a functional $F : (A \rightarrow B) \rightarrow C$ to a function $g : A \rightarrow B$ is realized as follows. $A \rightarrow B$ is the socket of the functional F . The application is done (see Fig. 4) by establishing appropriate directed connections (links). That is, the link between the socket A of the socket of F and the socket A of g , and the link between the plug B of g and the plug of the socket of F .

Composition functional (denoted by $compose_{A,B,C}$) for simple composition of two functions (the first function f of type $A \rightarrow B$, and the second one g of type $B \rightarrow C$) is realized as two boards with appropriate links shown in Fig. 5. It is easy to check (by following the links) that applying $compose_{A,B,C}$ to two functions (see Fig. 5) results in their composition.

Note that a higher order application (i.e. application of a functional to a function), and a functional for composition are constructed just by providing some links between sockets and plugs. Since link corresponds to synapse, it might be interesting whether these functionals have counterparts in the brain.

Each construction, like $F(g)$ and $compose_{A,B,C}(f; g)$, can be distinguished as an individual object (notion). Perhaps, in the brain, they correspond to concrete regions. This corresponds to a new paradigm called radical embodied neuroscience (REN), see Matyja and Dolega 2015 [24], Kiverstein and Miller 2015 [20].

Generally, discrimination of new notions by the human mind is crucial for reasoning. Once a notion is distinguished, it may be used in more sophisticated reasoning. This evidently

corresponds to the reflective abstraction introduced by Piaget, especially if the notions emerge as the results of constructions. Note that here constructions mean dynamic (re)configuration of links between sockets and plugs.

A functional of special interest is *Copy*. Once an object a is constructed, repeat the construction once again. So that $Copy(a)$ returns two object: the original a , and its copy a' . Although the meaning of *Copy* seems to be simple, its realization in the brain may be quite complex especially if the object a is of a higher order type.

If it is supposed that the construction of object a occupies some well defined region in the brain, then *Copy* may be realized by copying this region into a new “free region”. Since in Biology (living organisms) copying (procreation) is ubiquitous, let us take the implementation of the functional *Copy* as granted.

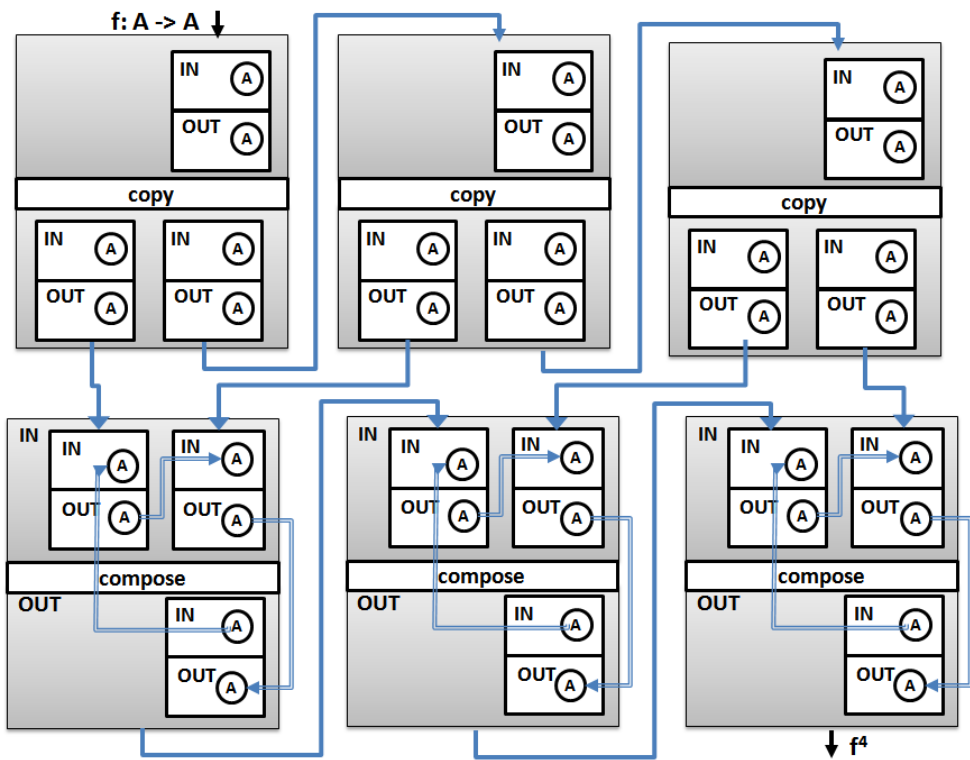


Figure 6. The result of application of the functional $Iter_A$ to natural number 4 and function f of type $A \rightarrow A$

Iteration as generalization of composition. That is, compose n -times a function $f : A \rightarrow A$ with itself. Note that n as a natural number is a parameter. The iteration is denoted by $Iter_A$ and it is a functional of type $(N; (A \rightarrow A)) \rightarrow (A \rightarrow A)$. So that $Iter_A(n; f)$ is the function being n -time composition of f . The realization of $Iter_A$ requires *Copy* for making copies of f , and $(n - 1)$ copies of the composition functional, see Fig. 6, where the

construction is done for n equal 4. Since natural numbers are involved in the functional, it seems that a hypothetical realization of *Iter*, in the brain, requires neurons.

Functional *Iter* is not the same as feedback loop that occurs when outputs of a circuit are routed back as inputs to the same circuit. The feedback enforces dynamics of the circuits, whereas *Iter* is static one shot operation.

Feedback loop can not be realized for higher order functionals where input as well as output are not electrical signals but higher order constructions.

Neural circuits are real dynamic systems where computation is done by consecutive processing signals (spike bursts). The circuits may be represented statically (without dynamics) as first order functions. Functionals are also static constructions operating on first order functions (circuits) by re(configuring) links inside and between the circuits.

Higher order primitive recursion schema (also known as Grzegorzczyk's iterator) can be constructed as a functional. For arbitrary type A , the iterator, denoted by R^A , of type $A \rightarrow ((N \rightarrow (A \rightarrow A)) \rightarrow (N \rightarrow A))$, is defined by the following equations.

For any $a : A$, $c : N \rightarrow (A \rightarrow A)$, and $k : N$

$$((R^A(a))(c))(1) = a \quad \text{and} \quad ((R^A(a))(c))(k + 1) = (c(k))(((R^A(a))(c))(k))$$

However, a construction of R^A does not follow from the definition. Actually, it is based on the iteration functional and consists on dynamic formation of links in boards of plugs and sockets. Higher order primitive recursion allows to define a large subclass of general recursive functions, e.g. the famous Ackerman function. This can be done on the basic level of the proposed approach to computable functionals. At higher levels of the approach (where functionals are used) all general recursive functions can be constructed. It seems that higher order computation involving the functionals is useful, especially as efficient and smart organizations of complex and sophisticated first order computations.

Note that there are next orders of constructions of functionals. Functionals operate on functionals (second order functions) are third order functions that operate on the second order functions by re(configuring) links in the boards of sockets and plugs. By analogy, this may be continued for the next higher orders constructions.

The complete presentation of the approach for constructing functionals is done in [3]. It is a strictly mathematical point of view.

Another point of view (close to the connectome) is considered in Section 4, where functionals are related to generic mechanisms for dynamic reconfigurations of compositions of first order functions. There, functionals are useful abstractions that reduce complexity of management of function compositions in large arrays of first order functions. Such array may be interpreted as a connectome structured by elementary neural circuits with connections between them that can be activated or deactivated.

3 Continuum

It seems that the notion of continuum has a straightforward and natural grounding in the human brain.

Vision sensory nervous system. The retina consists of about 130 million photo-receptor cells, however, there are roughly 1.2 million axons of ganglion cells that transmit information from the retina to the brain. It is interesting that a significant amount of visual pre-processing

is done between neurons in the retina. The axons form the optic nerve consisting of fibers (axons). Positions of the fibers in the nerve reflect the spatial and adjacent relations between the corresponding photo-receptors in the retina. In computations, the bundle of spikes in the nerve is considered together with the adjacent relation between the spikes. It is crucial for comprehending the notion of space Continuum.

The somatosensory system. Contrary to the vision system, it is spread through all major parts of mammal's body. Spatial and adjacent relations between nerve fibers of the somatosensory system contribute essentially to the notion of space Continuum.

The streams of spikes, in the nerve fibers delivered from sensory receptors to the brain, are not independent from each other; they are structured by causal and adjacent relations. The streams along with the relations are the grounding for the notion of space-time Continuum.

It is interesting that objects of the primitive types are based on neuron spikes. Natural number is just an isolated independent spike burst, whereas object of the type Continuum is a bundle of adjacent spike bursts.

A mathematical approach to the grounding (meaning) of the notion of Continuum is proposed in [4]. In the context of Neurobiology, of some interest may be also constructive foundations of Geometry in [6].

4 CGRA and connectome

A Coarse Grain Reconfigurable Architecture (CGRA) Bjorn De Sutter et al. (2013) [13], is a type of processor architecture that can be reconfigured at runtime. Array of interconnected functional units (first order functions implemented in hardware) can be dynamically configured (composed). Here, functionals may be envisioned as generic mechanisms for the management of connections (compositions) between the functions in the reconfigurable arrays. Hence, the concept of higher order computations as dynamic configuration of connections between hardware functional units is worth to be explored.

Reconfigurable computing architectures (Russell Tessier et al. (2015) [34]), and reconfigurable system (James Lyke et al. (2015) [22]) are active research subjects. However, the correspondence between functionals and generic mechanisms for dynamic reconfigurations is still not recognized in the hardware design.

Software and hardware are two different worlds. The first world consists of tightly coupled software (von Neumann programming languages) and extremely simple and ingenious von Neumann computing architecture. Computation is done on bytes sequentially on a single processor. The second world is hardware with its rich diversity of possible computing architectures, and possible grounding for higher order computations on functionals.

A new concept of programming (different from the von Neumann languages) is needed to shift the contemporary computing paradigm. Higher order computation can be envisioned as dynamic creation and reconfiguration of links between elementary circuits representing first order functions.

Perhaps the answer is in technology for developing large dynamically reconfigurable arrays of integrated circuits representing first order functions.

CGRA is an active research area. New concepts are explored like: data-flow graphs (Niedermeier et al. (2014) [29], and Palumbo et al. (2016) [30]), full pipelining and dynamic

composition (Cong et al. (2014) [12]), and overlay architecture (Capalija et al. (2014) [11], Jain et al. (2016) [17], and Andrews et al. (2016) [23]).

Functional units (FUs), as elementary first order functions, are built on fine grained integrated circuits. FUs are collected in an array, and connections between them are reconfigurable. A program (to be realized in hardware) is modeled as a data-flow graph where the nodes correspond to FUs. If the array is sufficient rich in FUs and possible connections, the graph can be mapped into the array. If the graphs are acyclic, then the data flow is fully pipelined, and can be realized in hardware, i.e. in the array of FUs.

Fully pipelined data-flows (as directed acyclic graphs) correspond to simple programs. Sophisticated programs may use recursion that presupposes dynamic unfolding, i.e. dynamic transformation of acyclic graphs, during execution, see Mukherjee et al. (2017) [26].

The transformation corresponds to a generic mechanisms for dynamic configuration of connections in large arrays of FUs. If the number of the connections is at most dozens (for simple computations), then dedicated mechanisms may be designed. However, if hundreds, thousands, and even more connections and FUs are needed, then the mechanisms must be generic, and must be based on higher order abstractions, i.e. higher order types and higher order functionals.

Hence, a functional is a fully pipelined data-flow (directed acyclic graph) with nodes corresponding to primitive functionals, and edges corresponding to flow of data from output of one node to the input of the next node. For complex functionals (with recursion) some nodes are dynamically unfolded.

From abstract mathematical point of view, functional is dynamic (depending on input values) transformation of acyclic directed graphs.

A functional is executable (is an executable program) if all input and output nodes of the graph are first order functions. This means that the functional (dynamically) configures compositions of the first order functions.

Executive programs can be compiled into hardware i.e. into an underlying array of FUs.

5 A non-von Neumann architecture (of human mind?)

Let us recall that functionals are constructed by dynamic creation and reconfiguration of links between sockets and plugs in the very similar way as the higher order organization of information processing in the neuronal connectome.

Links (connections) are between plugs and sockets of the same type. Although the type may be of higher order, they can be reduced to the links between sockets and plugs of the primitive type of natural numbers.

A link is always directed, i.e. it determines the direction of data flow.

We are going to propose a layered architecture of the human brain, see Fig. 7.

The basic layer is the connectome, i.e. neurons connected into circuits. It corresponds to hardware in the von Neumann computer architecture. It is 0-layer.

Let us suppose, for a moment, that the connectome can be represented as a repository (array) of elementary circuits corresponding to first order functions. The array is equipped with rich-clubs, silent synapses (that can be activated), and potential new synapses that can be created on demand. This constitutes the 1-st layer of the architecture.

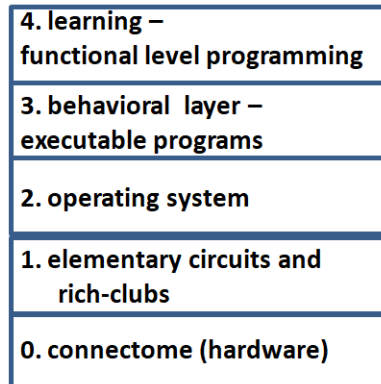


Figure 7. Layered architecture of a non von Neumann architecture (of human mind?)

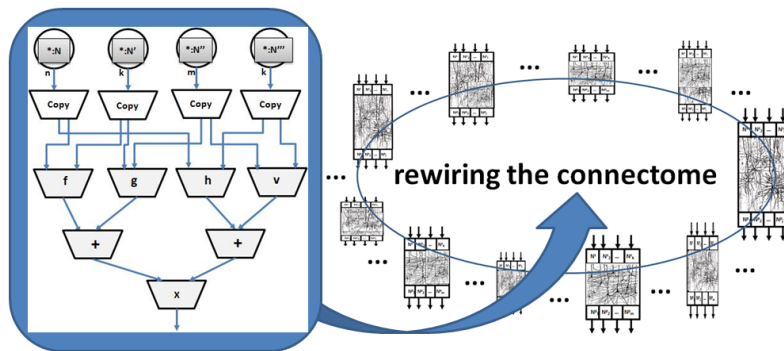


Figure 8. Executable functionals (programs) implemented in connectome

In order to compile an executive functional (program) into the repository (see Fig. 8), an operating system is needed for managing dynamic compositions of elementary neural circuits into a complex circuit, see Fig. 9. This constitutes the 2-nd layer of the architecture.

Executive programs constitute the 3-rd layer. It is called behavioral layer where, triggered by external (or internal) stimuli, executable programs are passed to the operating system (layer 2) to be compiled in layer 1 onto complex dynamic neural circuits.

Any such executive program is constructed from complex higher order functionals, the functionals that are constructed from primitive functionals according to generic rules. It is done in the 4-th layer called functional level programming layer. Construction of new functionals is related to learning in a general sense. Hence, the layer is called learning. This may be seen as higher order programming, i.e. constructions of higher order functionals. Hierarchy of such functionals corresponds to hierarchy of abstract notions.

Note that operating system, behavioral layer and learning are consecutive and hierarchical abstractions done above the physical layers of connectome with elementary circuits and rich-clubs. One may ask where and how these abstract layers are implemented in the brain.

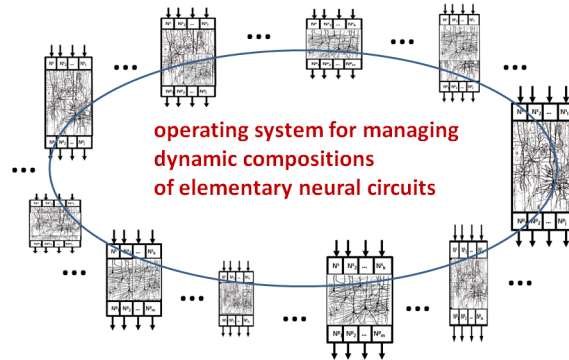


Figure 9. Operating system for managing dynamic reconfigurations of compositions of first order functions (i.e. elementary neural circuits)

6 Conclusion

It seems that there are two essential primitive types in Mathematics; the type of natural numbers, and the type of Continuum. Both types have their counterparts in the human brain. The natural numbers may be identified with individual and independent bursts of neuron spikes. The type Continuum has also the straightforward interpretation in the human brain. Sensory nervous systems support this view. The static interpretation of the neural circuits, as first order computable functions, seems to be justified. This may give rise to expect that higher order computable functions (functionals) have counterparts in the human brain.

It seems that the mechanisms (functionals) for rewiring the connectome (i.e. the meta-plasticity) may give rise to develop a non-von Neumann computer architecture and a corresponding function-level programming language postulated by John Backus 1977 [7]; for more on this subject see [3] and [5].

Finally, let me cite

- Johnson 2016 [18]: *Most non-von-Neumann architectures are modeled on the human brain. [...] "The future of AI will use humanlike sensors together with brain-like AI to make all sorts of ordinary devices smart," said Michael Mayberry, a corporate vice president and managing director of Intel Labs.*
- Bron 2023 [10]: *Today, neuromorphic computing stands alongside quantum computing, the Internet of Things, and artificial intelligence as one of the most transformative emerging technologies.*

References

1. Adolphs, R.: The unsolved problems of neuroscience. Trends in cognitive sciences **19**(4), 173–175 (2015)
2. Ambroszkiewicz, S.: On higher order computations and synaptic meta-plasticity in the human brain. In: International Conference on Artificial Neural Networks. pp. 145–152. Springer (2016)

3. Ambroszkiewicz, S.: Functionals and hardware. arxiv.org/abs/1501.03043 (2015), arxiv.org/abs/1501.03043
4. Ambroszkiewicz, S.: The grounding for continuum. arxiv.org/abs/1510.02787 (2015), <https://arxiv.org/abs/1510.02787>
5. Ambroszkiewicz, S.: On the notion of “von neumann vicious circle” coined by john backus. <http://arxiv.org/abs/1602.02715> (2016), <http://arxiv.org/abs/1602.02715>
6. Ambroszkiewicz, S.: Combinatorial constructions of intrinsic geometries. arxiv.org/abs/1904.05173 (2020), <https://arxiv.org/abs/1904.05173>
7. Backus, J.: Can programming be liberated from the von neumann style?: A functional style and its algebra of programs. *Commun. ACM* **21**(8), 613–641 (Aug 1978). <https://doi.org/10.1145/359576.359579>, <http://doi.acm.org/10.1145/359576.359579>
8. Bertolero, M.A., Yeo, B.T., D. Esposito, M.: The modular and integrative functional architecture of the human brain. *Proceedings of the National Academy of Sciences* **112**(49), E6798–E6807 (2015)
9. Braun, U., Schäfer, A., Walter, H., Erk, S., Romanczuk-Seiferth, N., Haddad, L., Schweiger, J.I., Grimm, O., Heinz, A., Tost, H., et al.: Dynamic reconfiguration of frontal brain networks during executive cognition in humans. *Proceedings of the National Academy of Sciences* **112**(37), 11678–11683 (2015)
10. Bron, D.: Brain Computers? . <https://medium.com/chain-reaction/brain-computers-976ae7ec19a8> (Oct 2023), <https://medium.com/chain-reaction/brain-computers-976ae7ec19a8>, medium.com
11. Capalija, D., Abdelrahman, T.S.: Tile-based bottom-up compilation of custom mesh-of-functional-units fpga overlays. In: 2014 24th International Conference on Field Programmable Logic and Applications (FPL). pp. 1–8. IEEE (2014)
12. Cong, J., Huang, H., Ma, C., Xiao, B., Zhou, P.: A fully pipelined and dynamically composable architecture of cgra. In: Field-Programmable Custom Computing Machines (FCCM), 2014 IEEE 22nd Annual International Symposium on. pp. 9–16. IEEE (2014)
13. De Sutter, B., Raghavan, P., Lambrechts, A.: Coarse-grained reconfigurable array architectures. In: *Handbook of signal processing systems*, pp. 553–592. Springer (2013)
14. Harris, J.A., Mihalas, S., Hirokawa, K.E., Whitesell, J.D., Choi, H., Bernard, A., Bohn, P., Caldejon, S., Casal, L., Cho, A., et al.: Hierarchical organization of cortical and thalamic connectivity. *Nature* **575**(7781), 195–202 (2019)
15. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences* **79**(8), 2554–2558 (1982)
16. Hopfield, J.J., Tank, D.W., et al.: Computing with neural circuits- a model. *Science* **233**(4764), 625–633 (1986)
17. Jain, A.K., Li, X., Fahmy, S.A., Maskell, D.L.: Adapting the dyser architecture with dsp blocks as an overlay for the xilinx zynq. *ACM SIGARCH Computer Architecture News* **43**(4), 28–33 (2016)
18. Johnson, R.C.: Non-von neumann computers providing brain-like functionality. *Communications of the ACM* (Nov 2016), <https://cacm.acm.org/news/non-von-neumann-computers-providing-brain-like-functionality/>
19. Kanold, P.O., Deng, R., Meng, X.: The integrative function of silent synapses on subplate neurons in cortical development and dysfunction. *Frontiers in Neuroanatomy* **13**, 41 (2019). <https://doi.org/10.3389/fnana.2019.00041>, <https://www.frontiersin.org/article/10.3389/fnana.2019.00041>
20. Kiverstein, J., Miller, M.: The embodied brain: Towards a radical embodied cognitive neuroscience. *Frontiers in Human Neuroscience* **9**(237) (2015). <https://doi.org/10.3389/fnhum.2015.00237>, http://www.frontiersin.org/human_neuroscience/10.3389/fnhum.2015.00237/abstract

21. Leopold, D.A., Strick, P.L., Bassett, D.S., Bruno, R.M., Cuntz, H., Harris, K.M., Oberlaender, M., Raichle, M.E.: Functional architecture of the cerebral cortex. In: *The Neocortex*, vol. 27, pp. 141–164. MIT Press (2019)
22. Lyke, J.C., Christodoulou, C.G., Vera, G.A., Edwards, A.H.: An introduction to reconfigurable systems. *Proceedings of the IEEE* **103**(3), 291–317 (2015)
23. Ma, S., Aklah, Z., Andrews, D.: Just in time assembly of accelerators. In: *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. pp. 173–178. ACM (2016)
24. Matyja, J.R., Dolega, K.: Radical embodied neuroscience - how and why? a commentary on: The embodied brain: Towards a radical embodied cognitive neuroscience, *front. hum. neurosci.* 06 may 2015, <http://dx.doi.org/10.3389/fnhum.2015.00237>. *Frontiers in Human Neuroscience* **9**(669) (2015). <https://doi.org/10.3389/fnhum.2015.00669>, http://www.frontiersin.org/human_neuroscience/10.3389/fnhum.2015.00669/full
25. McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics* **5**(4), 115–133 (1943)
26. Mukherjee, M., Fell, A., Guha, A.: Dfgentool: A dataflow graph generation tool for coarse grain reconfigurable architectures. In: *2017 30th International Conference on VLSI Design and 2017 16th International Conference on Embedded Systems (VLSID)*. pp. 67–72 (Jan 2017). <https://doi.org/10.1109/VLSID.2017.62>
27. von Neumann, J.: *The Computer and the Brain*. Yale University Press, New Haven, CT, USA (1958)
28. von Neumann, J., Burks, A.W., et al.: Theory of self-reproducing automata. *IEEE Transactions on Neural Networks* **5**(1), 3–14 (1966)
29. Niedermeier, A., Kuper, J., Smit, G.J.: A dataflow inspired programming paradigm for coarse-grained reconfigurable arrays. In: *International Symposium on Applied Reconfigurable Computing*. pp. 275–282. Springer (2014)
30. Palumbo, F., Sau, C., Fanni, T., Meloni, P., Raffo, L.: Dataflow-based design of coarse-grained reconfigurable platforms. In: *Signal Processing Systems (SiPS), 2016 IEEE International Workshop on*. pp. 127–129. IEEE (2016)
31. Park, H.J., Friston, K.: Structural and functional brain networks: from connections to cognition. *Science* **342**(6158), 1238411 (2013)
32. Sanz-Leon, P., Knock, S.A., Spiegler, A., Jirsa, V.K.: Mathematical framework for large-scale brain network modeling in the virtual brain. *NeuroImage* **111**, 385 – 430 (2015). <https://doi.org/http://dx.doi.org/10.1016/j.neuroimage.2015.01.002>, <http://www.sciencedirect.com/science/article/pii/S1053811915000051>
33. Siegelmann, H.T., Sontag, E.D.: On the computational power of neural nets. *Journal of computer and system sciences* **50**(1), 132–150 (1995)
34. Tessier, R., Pock, K., DeHon, A.: Reconfigurable computing architectures. *Proceedings of the IEEE* **103**(3), 332–354 (2015)