Theoretical Computer Science 777 (2019) 296–307

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Neural algebra on “how does the brain think?”

E. Engeler

ARTICLE INFO

Article history:
Received 21 May 2018
Received in revised form 19 November 2018
Accepted 27 March 2019
Available online 1 April 2019

Keywords:
Neural algebra
Neuro-algebraic thesis
Brain functions
Neural nets
Problem solving
Learning
Connectome
Attractor

ABSTRACT

The mathematical model employed in this essay attempts to explain how complex scripts of behaviour and conceptual content can reside in, combine and interact on large neural networks. The neural hypothesis attributes functions of the brain to sets of firing neurons dynamically: to sets of cascades of such firings, typically visualised by imaging technologies. Such sets are represented as the elements of what we call a neural algebra with their interaction as its basic operation. The neuro-algebraic thesis identifies “thoughts” with elements of a neural algebra and “thinking” with its basic operation.

We argue for this thesis by a thought-experiment. It examines examples of human thought processes in proposed emulation by neural algebras. In particular we analyse problems such as controlling, classifying and learning. In neural algebras these may be posed as algebraic equations, whose solutions may lead to extensions of a neural algebra by new elements. The modelling of such extensions consists of formal analogues to familiar faculties such as reflection, distinction and comprehension which will be mades precise as operations on the algebra.

An advantage of our approach is that this modelling leads directly to brain functions. From the cascades of such functions we obtain the neurons involved in them and their connective structure, and mathematically describe their behaviour.

Introduction

Challenged to answer the question in this title in five words or less, Steven Pinker replied “Brain cells fire in patterns.”[1] This strikingly adept answer was possible because by that time Crick’s *Neural Hypothesis*[2] was widely accepted. It states roughly that the neural activities of the brain may be all there is to the human mind. This had become a widely accepted presumption. Natural language, handy to a psychologist to formulate an answer, is not available to a mathematician: he his challenged to provide a formal theory of thoughts and a model of the thinking brain.

Well-known approaches for modelling brain functions have been around since the functional role of neurons was established and mathematicians such as Alan Turing, John von Neumann and Norbert Wiener started disciplines of mathematical theories of artificial neural nets and their possible role in the brain.

The present essay proposes *Neural Algebras* as mathematical models of the brain and advances the *Neuro-Algebraic Thesis* which identifies “thoughts” with elements of a neural algebra and “thinking” with composing its basic algebraic operation. Our *Thought Experiment* tests this hypothesis by constructing models of various mental activities and representing their interactions algebraically in a neural algebra. This modelling tacitly presupposes the validity of the Crick’s neural hypothesis.

E-mail address: erwin.engeler@gmail.com.

[1] Claire M. Guehenno, Pinker’s Brain picked On ’Colbert Report’. The Harvard Crimson, Febr. 8, 2007.

[2] Francis Crick, The Astonishing Hypothesis. Simon and Schuster, 1994.

https://doi.org/10.1016/j.tcs.2019.03.038
0304-3975/© 2019 Published by Elsevier B.V.

Neural algebras are super-structures above models of the brain. We view the "brain" as a directed graph of neurons that fire at discrete times. Each firing is induced according to a fixed law, specific to each neuron, by the firing of the neurons that are directed to it. Such firings produce cascades; the firing laws at each neuron govern the causations of these firings along paths in the graph. Sets of such cascades describe histories of firings, firing patterns. The firing laws lead directly to the definition of an operation between firing patterns by interpreting one pattern as acting on another at individual neurons, and thus to an algebraic structure on the set of firing patterns.

With this compositional interpretation on sets of cascades, or "brain functions" if you will, a brain model is furnished with an algebraic structure, which we call a *Neural Algebra*. This makes it possible to formulate questions about the functioning of the brain model through defining equations for chosen brain functions. In this sense, a neural algebra looks at the firing history of its neural net and explains it in its own chosen terms using equations, a familiar paradigm in the natural sciences.

An advantage of our approach is that this modelling operates directly on the neural correlates of the brain functions. From their cascades we arrive directly to the set of neurons involved in them and their connective structure. These constitute the *Connectome* of a brain function. We show that the solution to the defining equations reside in connectomes of distinct character and describe them and their behaviour mathematically. We shall also hazard to remark on some observable phenomena determined by the activities of such connectomes.

1. The brain as an algebra

Artificial neural nets may serve as brain models. Of these there is a large variety, including implementations in hard- and software, some in enormous installations.

The conceptually simplest model of a brain represents its connectivity, the *connectome* A, as a directed graph whose nodes, called neurons, fire at discrete time instances, understood as integers t in $\mathbb{Z}$. The global activity of the brain, the firing history of the brain, is given by the *firing function* $f(a,t)$ which takes the value 1 if the neuron a fires at time t and 0 otherwise. Modelling a brain is accomplished by imposing restrictions on the functions f by a *firing law*, inherited from abstracting neurological findings. The firing law at neuron b specifies the condition under which the firing of neurons $a_1,\dots,a_k$ at times $t_1,\dots,t_k$ causes the firing of neuron b at some later time t, assuming the former are connected to it by directed edges.

For example: In artificial neural nets a rudimentary firing law is based on assigning weights to the individual directed edges of the graph A: If the sum of weights of the incoming edges (synapses) exceeds a given threshold depending on the synapse, then the firing of the corresponding source neurons at time t causes the firing of the target neuron at time $t+1$. Positive weights correspond to excitatory, negative weights to inhibitory synapses.

The firing law at neuron b may be *dependent on time*; it may for example depend on the past history of the firings of b, e.g. for incorporating a regime of learning. Or it may result from some local or global inputs, thinking of clocks and of hormones. Such time-varying will be disregarded at first, but will be taken up in a later section.

The brain model $\mathcal{A}$

A directed graph A, together with a firing law $L(A)$ at all its nodes, constitute the connective basis of the brain model, denoted by $\mathcal{A}$. The model itself is built on this basis by identifying brain functions with parts of the firing history of $\mathcal{A}$ as follows:

Let T be a time interval, $T \subseteq \mathbb{Z}$. A firing function $f : A \times T \to \{0,1\}$ conforming to the firing law determines a *firing history* of a brain model based on $\mathcal{A}$. It may be visualised as a directed graph $H(\mathcal{A},T)$ whose nodes are labelled by $\langle a,t\rangle$ indicating the firing of neuron a at time t. The links of the graph connect firings $\langle a_1,t_1\rangle,\dots,\langle a_k,t_k\rangle$ with the firing $\langle b,t\rangle$ which is directly caused by the former according to the firing law at b. This law may distinguish individual synapses of b. In our notation, this distinction is represented by numerating the synapses of b as a sequence. The expression

$$[[\langle a_1,t_1\rangle, o, \langle a_2,t_2\rangle, o, \langle a_3,t_3\rangle], \langle b,t\rangle]$$

denotes the synaptic branching of the graph $H(\mathcal{A},T)$ at $\langle b,t\rangle$ with three occupied synapses and two unoccupied ones, (denoted by the letter o, mostly disregarded in the sequel.)

Cascades

A cascade is a part of the firing history that can be described as a subtree of $H(\mathcal{A},T)$. Using the above notation for synaptic branching, a tree-notation for cascades is obtained by recursion:

Individual nodes of $H(\mathcal{A},T)$ and o are cascades.

If $x_1,\dots,x_k$ are cascades with roots $\langle a_1,t_1\rangle,\dots,\langle a_k,t_k\rangle$, and these branch from a node $\langle b,t\rangle$, then $[[x_1,\dots,x_k],\langle b,t\rangle]$ is a cascade, assuming that the firing law at $\langle b,t\rangle$ determines that b fires at time t, provided that the neurons in $\langle a_1,t_1\rangle,\dots,\langle a_k,t_k\rangle$ fire at the indicated times $t_1,\dots t_k$ respectively. Unoccupied synapses are suppressed in this notation.

By definition, each cascade-tree is a causal history of firings: all its branchings are required to obey the firing laws. Note that the graph $H(\mathcal{A},T)$ may be understood as an overlapping union of its cascades; thus the set of cascades (redundantly) provides the complete history of firings.

Brain functions

We are going to use *sets of cascades*, the "firing patterns" of the introduction, as the basic objects of an algebra, a "neural algebra". They are intended to represent specific activities of a brain. The fact is, that such activities may influence each other. This can be understood as one brain activity operating on another brain activity to produce a brain activity. Some objects are therefore to be interpreted as functions, which means that its constituent cascades are to be viewed as input-output pairs.

A *Functional Interpretation of a Cascade* z selects a node $\langle c,t\rangle$ in z, which defines the functionality of z by its firing law as follows: Assume that $x_1,\dots,x_k$ are the sub-cascades of z with the same root $\langle c,t\rangle$ (the arguments), and let y be the remaining sub-cascade of z with $\langle c,t\rangle$ as one of its leaves, it is the characteristic leaf of this value. The cascade z with this interpretation is denoted by

$$[x_1,\dots x_k]\xrightarrow[c]{t} y.$$

The node $\langle c,t\rangle$ is called its *Key Node*; it is also the characteristic leave of y. – This is one functional interpretation of z, determined by the choice of $\langle c,t\rangle$ as its key node; there may be many others.

Brain Functions as sets of cascades

A set of cascades M acts functionally on its argument as follows: let N an arbitrary set of cascades. Then the operation of M on N results in a set of cascades $M\cdot N$ by the

Application Operation:

$M\cdot N=$ the set of all cascades y for which there exists a functional cascade $[x_1,\dots,x_k]\xrightarrow[c]{t} y$ in M such that $\{x_1,\dots,x_k\}$ is a subset of N.

Notation: In our notation this will read

$M\cdot N=\{y:\exists\beta\xrightarrow[c]{t} y\in M,\beta\subseteq N\}$, lower case greek letters are used whenever the individual x_i are not referred to.

Any set of cascades may be interpreted as a function by giving some of its element cascades a functional interpretation: For simplicity any set of cascades shall be called a *Brain Function*, independent whether it is functionally interpreted or not.

Remark: We suggest that if a brain function M operates on another brain function N, it may well be that N in turn operates on other brain functions. Viewed as a set of cascades, M would then contain output cascades that themselves have functional interpretations. Moreover, the input cascades of M also may also have functional interpretations: Our definition of a functional interpretation of a cascade admits functional cascades also for its component cascades. In this way we need not make distinctions of higher type level of brain functions (of brain functions, etc.).

This may remind the occasional reader of the type-free lambda calculus and combinatory logic. In fact, neural algebras are substructures of graph-models of combinatory logic.[3]

It might seem that some brain activities operate on more than one operand at a time, and that therefore we would have to consider functional interpretations of cascades with grouped branchings which reflect the distinct input operands. While easily formalised, this seems overly formalistic. Instead, we prefer to understand multiple operations such as $F(M,N)$ as $(F\cdot M)\cdot N$, as does combinatory logic, or sometimes more restrictively as $F\cdot M\cup F\cdot N$.

Neural Algebras

Let $\mathcal{A}$ be a brain model and $C(\mathcal{A},T)$ be its set of cascades, including functionally interpreted ones. To define a neural algebra we choose a collection $\mathcal{B}$ of subsets of $C(\mathcal{A},T)$. This collection of sets constitutes the set of elements of the algebra. With the application operation defined on $\mathcal{B}$ we have the makings of an algebraic structure, a

Neural Algebra.

A neural algebra $\mathcal{N}$ over a brain model $\mathcal{A}$ consists of a set $\mathcal{B}$ of brain functions, closed under the operations of application "$\cdot$" and union "$\cup$" and containing the empty set $\emptyset$. Symbolically: $\mathcal{N}=\langle\mathcal{B},\cdot,\cup\rangle$.

Taking the set $\mathcal{B}_\top$ of all subsets of $C(\mathcal{A},\mathbb{Z})$, together with all its possible functional interpretations, as its set of elements, we obtain the top element $\mathcal{N}_\top=\langle\mathcal{B}_\top,\cdot,\cup\rangle$ of the lattice of neural algebras over $\mathcal{A}$. We will mainly be interested in its subalgebras, their generation and their extensions.

If $\mathcal{G}$ is a set of subsets of $\mathcal{B}_\top$, then $\mathcal{N}(\mathcal{G})$ is the subalgebra of $\mathcal{N}_\top$ that is *generated* from $\mathcal{G}$ by the operations of application and finite union.

An *extension* $\mathcal{N}'$ of $\mathcal{N}$ inside $\mathcal{N}_\top$ results by adding generators to $\mathcal{N}$ and closing under application and union. An *algebraic extension* of $\mathcal{N}=\langle\mathcal{B},\cdot,\cup\rangle$ is generated by using as generators the set of solutions in $\mathcal{N}_\top$ of an equation with parameters in $\mathcal{B}$. Typical examples of such equations are the *fixpoint equation* $\varphi(X)=X$ and the *equilibrium equation* $\varphi(X)=\psi(X)$, where $\varphi(X)$ and $\psi(X)$ are algebraic expressions in $\cup$ and $\cdot$ with parameters in $\mathcal{B}$.

[3] In particular those in: E. Engeler, Algebras and Combinators, Algebra Universalis, 1981, 389–392.

The Thought Experiment
Are neural algebras faithful enough models of the human brain?
The background of the modelling is that the connective brain model $\mathcal{A}$ and the choice of $\mathcal{B}$ are supported by anatomical, physiological and other observational data based on actual brains and represent them faithfully. In this sense, a neural algebra is just one of many possible choices of sets of cascades and of functional interpretations of some of them, "explaining" the firing history of some actual brain.
Our thought experiment starts with the neural hypothesis: we consider mental activities such as problem solving, controlling a movement to obtain a goal, etc., and propose to model them as functionalities in the neural algebra. The task is to make the modelling persuasive.

To conclude, we would like to touch upon an important design aspect: The models $\mathcal{A}$ of the brain have a *topological character*: There is no geometrical location given for the individual neurons. In actual brains they are often grouped into specific modules. There is no indication in the model of the length of links, axons and dendrites, and their bundling between modules, save perhaps indirectly by the firing times. As I read the literature, it seems that these groupings are of central importance for their role as seats of brain functions. They may also differ in the ability to adapt the firing laws of individual neurons if this is included in the modelling.

2. Problem solving: persistent recursion and distinction

"All Life is Problem Solving". This statement, the title of a lecture given by Karl Popper,[4] is meant here to refer to the life of a brain, maybe the human brain, and perhaps to its many outside connections that help determine that life. What is a "Problem?" In lieu of a definition, we shall take recourse to examples. In neural algebra the conception of a problem is captured by its *Defining Equations* as illustrated in the following.

The first issue is to understand what kinds of sets of cascades we should reasonably consider.
The life span of the human brain is, sadly, finite. This is why we mentioned a finite interval T of $\mathbb{Z}$ in the definition of neural algebras. On the other hand, we are aware of different timescales: the timescale of cascades may be 10^2, of individual thoughts and actions it may easily be of the order of 10^5 firings; the whole life span is enormously larger. If we wish to talk about "thinking", it is therefore technically reasonable to set $T = \mathbb{Z} = \{\cdots - 2, -1, 0, 1, 2, \dots\}$ for the present and to restrict the modelling of some individual "thoughts" M to cascades of limited length. Moreover, for a set of finite cascades to be relevant and not only ephemeral in the neural algebra, it needs to have the stable presence of a distinct mental entity. This is the intent of the following definition:

Sustainment and Persistence.
If x is a cascade in $\mathcal{N}$ and $t \in \mathbb{Z}$ then the *time shift* of x to t, denoted by x^t, results from replacing the timing t_0 of the root of x by t and adapting all timings in the cascade x accordingly, (i.e. respecting the relays of the firing law at each node). For any set G of cascades and time instance t we let $G^t = \{x^t : x \in G\}$.
A brain function G is *sustained* if $G = \bigcup_{t\in\mathbb{Z}} {G_0}^t$ for some G_0. It is called *persistent modulo* G_0 if G_0 is finite.

Observe that for persistent G and H the application $G \cdot H$ is also persistent. To obtain its modulus, fix an arbitrary time instance s in $\mathbb{Z}$, let G_0, H_0 be moduli of G, H and let $H^s = \{x^s : x \in H_0\}$ where s is the latest timing of a root in H_0 and $G_s = \{\beta \xrightarrow[c]{s} x : \exists \beta \xrightarrow[c]{t} x \in G_0\}$. Then $G \cdot H = (\bigcup_{t\in\mathbb{Z}} {G_s}^t) \cdot H = \bigcup_{t\in\mathbb{Z}} ({G_s}^t \cdot H) = \bigcup_{t\in\mathbb{Z}} ({G_s}^t \cdot (H^s)^t) = \bigcup_{t\in\mathbb{Z}} (G_s \cdot H^s)^t$, and $G_s \cdot H^s$ is a modulus.
The same also holds for the union of two persistent sets: their union is persistent modulo the union of the respective moduli.
Notation. In the notation for functional cascades in persistent and sustained brain functions we drop the upper index t in the arrow. We also suppress the lower index c when it is clear from the context.

Control Problems.
The concept of *Control* involves a controlling operator C which acts on a controlled object X by application $C \cdot X$. The aim is to accomplish control, formulated as $C \cdot X = X$. If a brain is presented with a problem, it gathers into a brain function C all the faculties that may help in the solution. (The hypothesis does not distinguish between conscious and unconscious faculties.) Additionally, the control problem may come with the requirement that the solution process includes "getting control" of a given precondition X_0, which translates into $X_0 \subseteqq C \cdot X_0$, that is $X_0 \cup C \cdot X_0 = C \cdot X_0$. These two equations constitute the defining equations of the control problem.
Comment: With different kinds of initial conditions we could approach the modelling of concepts different from control such as *abstraction*, using $X_0 \supseteqq C \cdot X_0$, or *association*, when X_0 and $C \cdot X_0$ are incomparable. But this is another chapter.

What is a solution of a control problem? Let us look at examples:
All that is required is to find X that stabilises the control, $C \cdot X = X$. Such X may represent the history of control, e.g. X describes the trajectory of a ball thrown as X_0 under the control of the gravitational law C. The brain in essence incorporates a differential equation and solves it in the manner of the fixpoint existence proof.

[4] At Bad Homburg 1991, cf. K. Popper, All life if Problem Solving, Routledge 1999.

In an other case, the solution is obtained by *focusing* on the knowledge contained in X: To fix ideas, consider the problem of finding a real solution of $x^2 = 2$ with initial condition $0 < x < 1.5$ represented by the object X_0 of a neural algebra. For the process C the brain model may use the Sturm method of determining which of two half-segments of the one obtained before contains a root. It could declare the problem solved if for some given $\epsilon > 0$ the set X contains one of the finitely many necessary intervals (q_1, q_2) of this size compactly covering the original interval.
As an example closer to neuroscience, we mention the control circuits for the eye-movements to fix on an object; we shall return to this.

Formally, *Focusing* is a brain function F which has the character of a retraction, that is $F \cdot (F \cdot X) = F \cdot X$ for all X. It is typically obtained by specifying a list F_0 of acceptable forms of solutions (e.g. cascades representing the ϵ-intervals in the example above). Take $F = \{\sigma \xrightarrow[c]{t} x : x \in \sigma, \sigma \subseteq F_0\}$ then F retracts X to a solution set, possibly empty.

The Solution of the Control Problem and its Structure: Attractors

The *control equation*

$$C \cdot X = X, \text{ initial condition } X_0,\ X_0 \subseteqq C \cdot X_0$$

is a fixpoint equation which describes the fact that controlling X by a controller C succeeds in stabilising X starting with some initial X_0.
We now apply the result about the persistence of application to the control problem and consider its *control sequence* $X_0 \subseteq X_1 \subseteq X_2 \subseteq \dots$ determined by $X_{i+1} = C \cdot X_i, i \in \mathbb{N}$.
Then $C \cdot \bigcup_{i\in\mathbb{N}} X_i = \bigcup_{i\in\mathbb{N}} C \cdot X_i = \bigcup_{i\in\mathbb{N}} X_i$, that is $\bigcup_{i\in\mathbb{N}} X_i$ solves the control equation.
Since C and X_0 are persistent, all X_i are persistent, say modulo V_i respectively. Let ${V_i}^s$ be the *normalisation* of V_i, obtained through time-shifting V_i to a fixed time instance s. Note that ${V_i}^s \subseteq {V_{i+1}}^s$ and, since all ${V_i}^s$ are finite subsets of $\{y : \exists\beta \rightarrow y \in C_s\}$, their sequence becomes stable, say with ${V_m}^s$. Therefore all later X_i, being persistent modulo V_m, are equal, and X_m is in fact a solution of the control equation and persistent.
The recursion operator Γ denotes the extension of a neural algebra by the solution of a control equation $C \cdot X = X$ with initial condition X_0. The extension is denoted by $\Gamma_X(C, X)$. (The notation Γ_X binds the argument X.) The new generator is $\Gamma_X(C, X) \cdot X_0$.
In the case of persistent C and X_0 this generator is the above, and persistent, namely X_m with modulus V_m. Borrowing a notion from system theory, we may call this solution process an *attraction*. If V_m is nonempty we would call it an *attractor*. Note that the brain needs some time to obtain this desired object.

The Connectome of an Attraction

The connectome of a brain function M is a subgraph of the connectome A of $\mathcal{N}$. The connectome of an attraction determined by a recursion Γ is that part of the connectome A of $\mathcal{N}$ in which the control sequence resides, the connectome of the resulting modulus V_m of the recursion. It emerges from the proof of its existence as a set of nested cycles:
Consider the set of graphs of cascades in V_m and relabel the nodes by the corresponding neurons, suppressing time indices. Then identify each root with all the neural nodes in all the cascades with the same label. Unused leaves are firing continuously as "inputs", unused roots are "outputs. The resulting connectome consists therefore of directed cycles (and cycles within cycles, ...) with some adjoining and some departing links.

Remarks

Observables: Returning to the general description of problem solving that led to the notion of a control problem at the beginning of this section, we note that the activity in the connectome of attraction described above illustrates the intuition that arises from observing brain activities in human problem solving. V_m, the "result" of the control problem, has the appearance of *Coherence or Synchronicity* if the attractor covers a larger region of the brain, especially if it links different regions. One can also observe *Periodicities*, for example with period $t_{k+1} - t_k$ for all $t_k \geq t_m$, the time of stabilisation. The individual cycles in the attractor have their own periodicities, factoring into the whole. These may be considered distinct parts of the solution and may reside in distinct brain contexts. They are conceivably what is observed in the graphs of electroencephalograms.

Composite Control: We have shown above that algebraic composites $\varphi(A_1, \dots A_k)$ of persistent objects A_i are persistent. Hence the above results also apply to equations $\varphi(A_1, \dots A_k) \cdot X = X$, and control may be the result of collaboration of a group of individual modules that a discernible by a contextual analysis of distinct brain functions. Some of these may represented by persistent objects of the nature of routine reactions, a fixed memory, etc., and are therefore modelled by persistent brain functions. For example:
"Flight upon a threat", proposing a somewhat naive example, describes control of his behaviour X by a little rabbit R. This is modelled by the equation

$$(R \cdot (B \cdot V \cup F \cdot V \cup E)) \cdot X = X.$$

The rabbit R uses its brain function B to judge the size of the threatening foe as seen by the visual input V. Using his brain function F on this input it judges how fast the foe seems to approach. By remembering knowledge E about the environment it judges how unsave the environment seems, all modelled by persistent objects.

Reflection: Persistent objects can also apply to themselves, resulting in persistent objects. Consider the equation

$$F \cdot F = F, \text{ initial condition } F_0.$$

Expanding F_0 by a solution is called a *Reflection* on F_0; it is an attraction.

Homoeostasis: The control obtained by using some persistent controlling object (or as a composition of such objects) applied to the controlled object X as above is only a first approximation to cybernetics[5] within neural algebra. It does not incorporate the important biological notion of *homeostasis*, of which the following equation is a simple example

$$(C \cdot (P \cdot X \cup E)) \cdot X = X.$$

The controlling object bases its functioning on its present perception P of the controlled object X, together with a memory E of exceptional cases of control conditions. Thus, P could for example determine how far the present controlled object X is from a desired objective. This is representative for homeostasis. Our results on attraction and attractors of course also apply to the more general equational problems of this form, that is for equations of the form $\varphi(A_1, \dots, A_n, X) = X$ with persistent A_i and X.

The objects P, C, E may well be sustained, but in fact they may be time dependent, interpreting E as describing not a persistent memory of exceptional cases but a time-varying influence from a changing environment. Therefore, a more general control equation needs not only to include X in the control expression but admit as parameters arbitrary time-varying objects $U_1, U_2, \dots$ expressed as $\varphi(X, U_1, \dots U_k) = X$.

Discrimination

In the course of problem solving in general, we may encounter the question which of two, or any number of, alternative faculties (for example of focusing), we should employ. In the case of control, this means to decide between control operators. To make distinctions between objects and take decisions accordingly is one of the basic faculties of the mind.

The operation of distinction is based on two brain functions D and D', *distinction criteria* on which we make the choice between two actions A and B on an input X by choosing A if $D \cdot X = \mathbf{y}$, and B if $D' \cdot X = \mathbf{y}$. This $\mathbf{y}$ is the persistent closure $\{{y_0}^t : t \in \mathbf{Z}\}$ of a special cascade y_0 chosen for the purpose. (In section 4, we shall choose the number object $\mathbf{0}$ for that purpose.)

Note that D', the opposite of D, need not be its complement; in that case the distinction operator may return the empty set for some X as $D' \cdot X = \emptyset$ and $D' \cdot X = \emptyset$, representing inability to choose. Of course, D' may in cases be construed as the complement.

The discrimination operator Δ is defined by:

$$\begin{aligned}\Delta(X; D, D'; A, B) = &\{\beta \to (\alpha \to x) : \alpha, \beta \subseteq X, \alpha \to x \in A, \beta \to y_0 \in D\} \\ &\cup \{\beta \to (\alpha \to x) : \alpha, \beta \subseteq X, \alpha \to x \in A, \beta \to y_0 \in D'\}\end{aligned}$$

Observe that

$$\Delta(X; D, D'; A, B) \cdot X = A \cdot X \text{ if } D \cdot X = \mathbf{y},$$
$$\Delta(X; D, D'; A, B) \cdot X = B \cdot X \text{ if } D' \cdot X = \mathbf{y},$$
$$\Delta(X; D, D'; A, B) \cdot X = \emptyset \text{ otherwise}.$$

If all the parameters in Δ are persistent, respectively sustained, then so is each of its two components.

For decisions with more than one argument, say X and Y, the criteria would apply as $D \cdot X \cdot Y$ and $D' \cdot X \cdot Y$ and we would write $\Delta(X, Y; D, D'; A, B)$.

Simultaneous and Joint Control

It happens often in the life of a mind that two interrelated problems present themselves at the same time. If the interrelation is explicitly given by two objects controlling the solution processes for each of the problems, we have the easier case of simultaneous control. If there is only one problem to be solved, but we have to balance between two independent control regimes, we arrive at an equilibrium equation.

[5] N. Wiener, Cybernetics, MIT Press, 1948. It was to an important extent motivated by trying to develop mathematical models of control by the brain.

Simultaneous control in its simplest form is given by two simultaneous equations $A \cdot X \cdot Y = X$ and $B \cdot X \cdot Y = Y$ with persistent A, B. The initial conditions are of course $X_0 \subseteq A \cdot X_0 \cdot Y_0$ and $Y_0 \subseteq B \cdot X_0 \cdot Y_0$. The solution algorithm is the same as above, except that it employs two procedures simultaneously: $X_{i+1} = A \cdot X_i \cdot Y_i$ and $Y_{i+1} = B \cdot X_i \cdot Y_i$.

Equilibrium equations $\varphi(X) = \psi(X)$ have the shape of equations familiar from the physical sciences; remember the partial differential equations of physics. Joint control $A \cdot X = B \cdot X$ is the simplest form of the equilibrium equation. The initial conditions are simply $X_0 \subseteq A \cdot X_0$ and $X_0 \subseteq B \cdot X_0$. The control operators A and B as well as X_0 are assumed persistent. Example: Riding a bicycle while trying to keep A the balance and follow B the road around a curve: $A \cdot X = B \cdot X$.

Given A and B, to find a non-trivial solution we may try to institute a sequence $X_0, X_1, \dots$ of approximations, persistent sets with moduli $V_0, V_1, \dots$. To generate the sequence, we proceed on the pattern of recursion, except that now we alternate composition with A and B. Computing $X_{i+1} = A \cdot X_i$ and $X_{i+1} = B \cdot X_i$, one of these may be included in the other; this is the one we choose for X_{i+1} and then continue. However, the two may be incomparable, which means that the process cannot succeed in a solution; X_i is hopeless. We are led to the familiar backtrack procedure: we proceed to guessing some extension $X_i{}'$ at the earlier stage and try again to go on to $A \cdot X_i{}'$ and $B \cdot X_i{}'$, etc. In fact, we may have to try finitely many $X_i{}'$ (by choosing among the finitely many possible moduli $V_i{}' \supseteq V_{i-1}$ determined by A, B.) If this again leads to incomparable sets, we would have to backtrack further, conceivably reaching X_0 and then $\emptyset$. If not, the sequence $X_0, X_1, \dots$ stabilises in a set X_m as in the recursion case with a result that is of the same form of attractors. Again, the algorithm describes a familiar story, it either finds a solution or terminates with finding that there is none. The weaving bicycle ride of a beginner illustrates the solution process.

The algorithms above of course generalise to simultaneous control equations of the form $\varphi(A_1, \dots, A_n, X, Y) = X$, $\psi(B_1, \dots, B_m, X, Y) = Y$ and to equilibrium equations of the form $\varphi(A_1, \dots, A_n, X, Y) = \psi(B_1, \dots, B_m, X, Y)$ with persistent parameters and initial conditions X_0, Y_0. This allows to treat second-, higher-order and self-aware control, to modelling forms of adaptation, various regimes of governing and others, some still to be developed. For example, it can be shown[6] that any finite set of multi-variable equations can be transformed into an equation of the form $A \cdot X = B \cdot X$ whose solutions can be effectively transported to the original equations.

Question: The equilibrium algorithm succeeds because the search space for the connectome of a possible solution is finite, determined by the moduli of A and B. But for the general equilibrium equation where the parameters are allowed to be only subsistent: Can there be a general algorithm in this case. *No !* as shown in section 4.

Remark: Who is the "we" in the description of the solution processes above? In the context of this paper we expect it to be a functional element of the neural algebra that contains the parameters, or an extension thereof. In section 4 we shall demonstrate the existence of such extensions by modelling algorithms using partial recursive functions. We shall argue there that the Church-Turing Thesis applies to all "rich enough" neural algebras: any effective algorithm on persistent elements is representable by a sustained functional element.

3. Development by learning and comprehending

Learning.

For a neural algebra $\mathcal{N}$, to develop means to expand by some new generators which $\mathcal{N}$ has "learned" through some process, e.g. in the course of solving a control problem. The ability to learn is surely one of the most prominent faculties of the brain. Learning to control an object through the process of solving a simple control equation $C \cdot P = P$ means to "pay attention" to the solution process. The moduli $P_1, P_2, \dots,$ starting from the initial condition P_0 make their successive appearances. P_0 *appears* at a certain time t_0, the timing of the latest root of its modulus, and P_k has a latest root t_k. Of course, if P_{k+1} happens to equal P_k then C is good enough, the learner has obtained a solution proposal – if it can wait that long, (and the original control C is good enough, otherwise the learner should try to find a new C).

The *Attention Span* of the learner is a time limit t_l which enters the definition of attention by determining when the sequence P_k looses attention, namely when $t_k \geq t_l$. Then $\Lambda(t_l, C, P_0) = P_l$ is the outcome of learning the control equation for a learner with attention span t_l. It may not be a solution.

For a different model of attention we could make use of the "plasticity" of $\mathcal{N}$ by allowing the firing law of individual neurons to depend on their firing history. This opens up a whole new area of learning algorithms, some of which much used in industry. But this is an other chapter.

Teaching

Supervised learning may be modelled by a system of *simultaneous fixpoint equations* with persistent parameters.

Example: Consider the following, admittedly superficial, description of *learning to play the violin*. Construct again a control equation; but now enters the *teacher* T and a piece P that he wishes the student to learn. Accordingly, the teacher adapts his teaching of the student C into a teaching operation $T \cdot P$. The teaching equations are the simultaneous control equations

[6] By an effective construction illustrated in E. Engeler, Equations in combinatory algebras. LNCS 164, Springer 1984, on graph models.

$$(T \cdot P) \cdot C = C, \quad C \cdot X = X.$$

Teaching starts with an initial performance X_0, imagine some scratching on the violin by the initial control C_0 of the violin by an inexperienced student. Assuming T and P to be persistent, we are assured of a good learning situation as above, provided the attention span of teacher and student suffice. Of course, the result still has to be tested.

Scholium

Perhaps there is a lesson in this.

Consider how the eye moves to focus on an object. The control C changes the vector X between object and present focus in the eye field until it is small enough, using visual inputs and input/output from motor neurons. This is an example of a neural control mechanism that has been studied in detail for the mouse brain.[7] The connectome of the control uses a considerable number of neurons, grouped in modules and connected by excitatory and inhibitory synapses. It consists of an intricate pattern of nested loops going through five layers of the cortex. – The point here is: This control C has been obtained, learned, through innumerable generations, from eucaryotes to mammals, exposed to a world W in order to arrive at a solution of the equation $W \cdot C = C$. So, the control of the eye movement is an attraction, and it is really no surprise that attractor connectomes are ubiquitous in the cortex (and in "deep learning" artificial neural nets).

Comprehension

There is an aspect, for the violinist, from which we may derive another mode of extension of a neural algebra: – No amount of rote teaching will turn the student into a concert soloist. The sort of teaching above resembles more teaching new tricks to a dog. Artistic development is more than that, it involves aspects such as interaction with the public L and its reaction to the repertoire P. Apart from the influence of the teacher it also would include the understanding of the history H of the piece and its past interpretations I by other violinists. By the neuro-algebraic thesis each of these aspects, as a module in the connectome, is represented as an object of $\mathcal{N}$. The development of playing thus depends on an algebraic combination φ of these influences. The corresponding control equation takes on the form

$$\varphi(C, P, L, H, I) \cdot X = X$$

which includes understanding, "*comprendre*" in French, of the musical culture around the violinist. The expression $\varphi(C, P, L, H, I)$ may be quite involved.

In terms of brain functions, to understand means to have available, by "comprehension", a new brain function R such that (with parentheses to the left)

$$R \cdot C \cdot P \cdot L \cdot H \cdot I = \varphi(C, P, L, H, I).$$

This may be considered an example of a general principle:

Principle of Comprehension

Let $\mathcal{N} = \langle \mathcal{B}, \cdot, \cup \rangle$ be a neural algebra and let $\varphi(A_1, \ldots, A_n)$ be an algebraic expression composed of persistent objects $A_1, \ldots, A_n \in \mathcal{B}$ by means of the application operation and finite union. Then there is an algebraic extension of $\mathcal{N}$ by a persistent generator $R = \Pi\varphi(A_1 \ldots, A_n)$ which solves the equation

$$R \cdot A_1 \cdot A_2 \cdot \cdots \cdot A_n = \varphi(A_1, \ldots, A_n).$$

Notation: Using lower case greek letters for denoting arguments in functional cascades we write $\alpha \to x$ for $\alpha \xrightarrow[c]{t} x$, suppressing the mention of neuron c, and of time t. The elements of R thus have the form $\alpha_1 \to (\alpha_2 \to (\ldots \to (\alpha_n \to x))\ldots)$.

Consider first evaluations of applications only; then the proof of this principle consists of straightforward repeated use of the definition of the application operation as illustrated in the following example.

$$\begin{aligned}
&A \cdot (B \cdot C) = \\
&= \{x : \exists \beta \subseteq B \cdot C, \beta \to x \in A\}, \text{with } \beta = [b_1, \ldots, b_n] \\
&= \{x : \exists \gamma_1, \ldots, \gamma_n \subseteq C, \exists b_1 \ldots b_n,\ [\gamma_1 \to b_1, \ldots, \gamma_n \to b_n] \in B, [b_1, \ldots, b_n] \to x \in A\}.
\end{aligned}$$

Hence, take

$$\begin{aligned}
&R = \\
&= \{[[b_1 \ldots, b_n] \to x] \to ([\gamma_1 \to b_1, \ldots, \gamma_n \to b_n] \to ([\gamma_1 \ldots, \gamma_n] \to x)) : \\
&\exists n, \exists [b_1 \ldots, b_n] \to x \in A, \exists \gamma_1, \ldots, \gamma_n \subseteq C, \gamma_1 \to b_1, \ldots, \gamma_n \to b_n \in B\}.
\end{aligned}$$

[7] Heinzle, J., Hepp, K., Martin, K.A.C., J. Neurosc. 27 (2007), 9341–9353.

To include the operation of union we take

$$R = \{\alpha \to (\beta \to x) : x \in \alpha \cup \beta, \alpha \subseteq A, \beta \subseteq B\}$$

to obtain $R \cdot A \cdot B = A \cup B$.

Remark: In order to construct $\Pi\varphi(A_1 \ldots, A_n)$ we needed to introduce new connections and new neurons (whose mention as subscripts to the arrows we suppressed). In particular, to "comprehend" $A \cdot (B \cdot C)$ there arose the need to develop insight into the structure of A, B and C, which means to recognise the cascades $\gamma_i \to b_i$. This is the "development" that has to be accomplished for "understanding" $A \cdot (B \cdot C)$. Speaking loosely about the human mind needing to do this, we might see a graduation of its "power of comprehension" in the amount of nesting of $\varphi(A_1 \ldots, A_n)$ that it can handle. Without question, there are great differences in the strengths of minds. For example, one may look at the mind of the self-avowed genius Hercule Poirot in "Five Little Pigs" by Agatha Christie. In the final scene of this delightful detective story, Poirot presents his parsing of the events that led to the trial and conviction of the wrong person: There are five people P, C, Y, E, A and two relevant facts F_1, F_2. The story combines them as follows:

$$P \cdot (P \cdot (((C \cdot F_1) \cdot ((C \cdot A) \cdot (C \cdot A))) \cdot (C \cdot (A \cdot Y)))) \cdot ((P \cdot F_2) \cdot (E \cdot Y))$$

The combinatory intelligence of Poirot, P, allows him to combine this into one convincing concept by which he exposed the real murderess E of Y. It is a fascinating that readers of detective stories can follow such considerable combinatory complexity with their own comprehension – and enjoy it.

4. The development of faculties by concept formation

Leaving violinist, teachers and dogs aside and turning to the mathematician: It is here that we find our most transparent examples of the extension of neural algebras. Consider for example the concept of prime number. A number p is prime if the algorithm of trying to find divisors of p terminates with only the number one. This concept is mathematically based on, composed of, other concepts such as divisibility and algorithm and therefore needs a "good enough brain", that is a neural algebra, to contain such objects and with the right algebraic properties. Fortunately, such a construction is textbook material,[8] where it is shown that the objects and results of recursion theory can be represented in a graph model of combinatory logic. This is inherited by neural algebras which are a special form of such models. We now sketch this representation.

Recursion Theory

The objects of recursion theory are the natural numbers and the partial recursive functions. These are to be modelled as elements in a neural algebra, the *recursionist* $\mathcal{N}_{\text{rec}}$.

Individual *natural numbers* are represented by persistent objects $\mathbf{0}, \mathbf{1}, \mathbf{2}, \ldots$, for example $\mathbf{n} = \{\langle n, t\rangle : t \in \mathbb{Z}\}$, where $0, 1, 2, \cdots \in \mathbb{N}$ are the neurons used for these numbers.

The successor function $s(n) = n + 1$ is the represented by

$$\mathbf{s} = \{[x] \xrightarrow[s]{t} y : x \in \mathbf{n}, y \in \mathbf{n+1},\ n \in \mathbb{N}, t \in \mathbb{Z}\}.$$

This yields $\mathbf{s} \cdot \mathbf{n} = \mathbf{n+1}$. The successor function is one of the basic functions of recursion theory. In general a partial function $f : \mathbb{N}^k \to \mathbb{N}$ is *represented* by an object $\mathbf{f}$ in $\mathcal{N}_{\text{rec}}$ if $f(n_1, \ldots, n_k) = m$ is defined and equal to m if and only if $\mathbf{f n}_1 \mathbf{n}_2 \cdot \cdots \cdot \mathbf{n}_k = \mathbf{m}$.

Notation: We continue to drop the application sign "·" as well as parentheses to the left and to use **boldface** to denote the representing objects as we encounter them.

For the model mathematician, the *recursionist* $\mathcal{N}_{\text{rec}}$, we have to provide generators to construct all partial recursive functions by starting from basic functions such as **s** and using expansions by reflection, comprehension and distinction on already constructed objects.

In addition to **s** we also use **p**, the predecessor function $p(n+1) = n, n \in \mathbb{N}$, and **z**, the zero function $z(n) = 0, n \in \mathbb{N}$, represented respectively by

$$\mathbf{p} = \{[x] \xrightarrow[s]{t} y : x \in \mathbf{n+1}, y \in \mathbf{n},\ n \in \mathbb{N}, t \in \mathbb{Z}\},$$

$$\mathbf{z} = \{[x] \xrightarrow[z]{t} y : x \in \mathbf{n}, y \in \mathbf{0}, n \in \mathbb{N}, n \neq 0, t \in \mathbb{Z}\},\ (\text{noting } \mathbf{z} \cdot \mathbf{0} = \emptyset),$$

[8] E.g. E. Engeler, Foundations of Mathematics: Questions of Analysis, Geometry, and Algorithmics, Springer Verlag 1993, Chapter 3; also in Russian, Chinese and German.

as well as the *Projection Functions* $u^i_m(n_1,\dots,n_m) = n_i$ defined using comprehension for $\varphi(\mathbf{n_1}\dots\mathbf{n}_m) = \mathbf{n_i}$. We obtain $\mathbf{u}^i_m\mathbf{n_1}\dots\mathbf{n}_m = \Pi\varphi(\mathbf{n_1}\dots\mathbf{n}_m)\mathbf{n_1}\dots\mathbf{n}_m = \mathbf{n_i}$.

Conceptual Objects

It is important to note that $\mathbf{p}, \mathbf{s}, \mathbf{z}$ and $\mathbf{u^i_m}$, being infinitary, are *not persistent but only sustained*. We call such brain functions *conceptual*. The recursionist deals mostly with conceptual brain functions.

Some results about persistent brain functions generalise to conceptual ones:

If $A_1,\dots A_k$ are conceptual then so is any algebraic $\varphi(A_1,\dots A_k)$.

The control equation $C \cdot X = X$ with conceptual C and conceptual initial condition X_0 has a conceptual solution, $\Gamma_X(C,X)\cdot X_0$. This holds also for $C = \varphi(X, A_1,\dots A_k)$ for conceptual $A_1,\dots A_k$. The expansion operations Δ and Π also preserve conceptuality.

Remark: We may ask how $\mathbf{s}$ and other conceptual objects are to be visualised as connectomes, i.e. as subgraphs of A. To consider a simple case, take the successor function, represented by $\mathbf{s}$ which is infinite and may be visualised in $\mathcal{A}$ as a helix consisting of the number objects $\mathbf{0}, \mathbf{1}, \mathbf{2}, \dots$ along successive cascades $[\mathbf{n}] \underset{s}{\to} \mathbf{n}+\mathbf{1}$ winding along an infinite sequence of copies of the node s. These nodes are then identified: the helical loops are glued at s to form a coiled infinite helical spring in A.

Partial Recursive Functions

These functions are generated starting from the basic functions by using composition, primitive recursion and the μ-Scheme. These constructions can be represented in $\mathcal{N}_{\text{rec}}$ by using the extension operations Γ, Δ and Π. As this construction progresses, we will find that not all function objects $\mathbf{f}$ that will be generated return numerical values $\mathbf{fn}$ for numerical input $\mathbf{n}$; the return may be the empty set.

Composition of functions such as

$$f(x_1,\dots,x_n) = h(g_1(x_1,\dots,x_n),\dots,g_m(x_1,\dots,x_n))$$

is represented by $\mathbf{hg}_1(\mathbf{k}_1\mathbf{k}_2\dots\mathbf{k}_n)\dots\mathbf{g}_m(\mathbf{k}_1\mathbf{k}_2\dots\mathbf{k}_n)$.

Primitive Recursion is defined by

$f(n) = \texttt{if}\, n = 0\ \texttt{then}\ k\ \texttt{else}\ g(f(n-1), n-1)$, with given function g. Its representation is obtained as follows:

Assume that the function g is represented by the conceptual object $\mathbf{g}$ such that $g(n,m) = k$ iff $\mathbf{g}\cdot\mathbf{n}\cdot\mathbf{m} = \mathbf{k}$. Using $\mathbf{g}$, the conceptual element $\mathbf{f}$ is defined by using the discrimination operator on the criterion of being $\mathbf{0}$ or being $\mathbf{n}+\mathbf{1}$ for some n. The distinction operators are therefore D, the union of all $\{\alpha \to \langle 0,0\rangle : \langle 0,0\rangle \in \alpha\}^t$, and D' correspondingly with $\langle 0,0\rangle \notin \alpha$. Constructing A and B such that $A\cdot X = \mathbf{s}^k \cdot X$, $B\cdot X = \mathbf{g}(\mathbf{p}X)$, we have

$$\mathbf{f}\cdot X = \Delta(X, D, D', A, B)\cdot X.$$

The μ-Scheme assumes a function g(x,y) and defines a new function by $f(x) = \mu y(g(x,y) = 0)$, that is the smallest y, if it exists, for which $g(x,y)$ is defined and equal zero. The construction of a representing object $\mathbf{f}$ for this function is similar to the one above.

Altogether, our construction maps all partial recursive functions into representing objects in a suitably rich $\mathcal{N}_{\text{rec}}$. As a consequence, we propose the following consideration:

Unsolvability

The question at the end of section 2 was whether it would be possible to design an algorithm to solve in general the equilibrium problem $\varphi(A_1,\dots,A_n,X) = \psi(B_1,\dots,B_m,X)$ with non-persistent, but sustained parameters.

Now that we have partial recursive functions available, recursion theory produces unsolvable examples, functions f and g such that there is no algorithm for finding $\mathbf{n}$ such that $\mathbf{f}\cdot\mathbf{n} = \mathbf{g}\cdot\mathbf{n}$.[9] Such a putative algorithm would reside in an expansion $\mathcal{N}_{meta}$ of $\mathcal{N}_{rec}$ to a *meta-level*, resulting in the *Gödel extension* generated by Gödel numbering, an assumption that leads to contradiction.

Gödel numbering is a primitive recursive algorithm, *on the meta-level*. It uses the recursive generation of all partial recursive functions (as described above) to enumerate them as $g_0, g_1, \dots$. In this way each such function g_j obtains its Gödel number $\ddot{g}_j$. There is also an inverse meta-primitive recursive algorithm which recoups g_j from $\ddot{g}_j$. By the famous Church-Turing Thesis, these algorithms can be made explicit as recursive functions and therefore are representable in the Gödel extension $\mathcal{N}_{meta}$.

[9] See loc. cit. in footnote 8.

5. Algorithmic faculties

The neural hypothesis claims that the faculties of the mind are realised by functions in the brain. The *Neuro-algebraic Thesis* claims that this can be realised by objects in a neural algebra. The challenge to support this thesis is met here by some thought experiments. It even leads to some predictions about structure and behaviour in problem solving and learning. We modelled the development of the brain with extensions of neural algebras, including discrimination, reflection and comprehension, each of them an algorithm on persistent objects, exemplified on a procession of virtual rabbits, dogs, violinists, mathematicians and metamathematicians. Such a sequence of model extensions can be viewed as the development of a *mind*. In actual brains much of this seems to be generated by exposure to external inputs, consequences of internal operations on native functionalities, some quite random.

In the development of the virtual minds illustrated above, four important faculties of the mind were identified as objects of neural algebra:

$\mathbf{\Gamma}$, the faculty of posing a problem by specifying a range of possible solution paths C and a seed object X_0 from which the solution X is to be obtained and made into a stable object: *Recursion*.

$\mathbf{\Pi}$, the faculty of perceiving the combinatory dependence-structure of a collection $A_1, \ldots, A_n$ of concepts and combine them: *Comprehension*.

$\mathbf{\Delta}$, the faculty of discerning criteria D, D' upon which to base the choice between two concepts A, B to be pursued: *Discrimination*.

$\mathbf{\Lambda}$, the faculty to control the solution of a problem and to fix on a solution: *Learning*.

These faculties are in effect algorithms on sets of cascades which in this essay have been made explicit to varying extent.

Concerning algorithms We still have to present the argument for the *neuro-algebraic Church-Turing Thesis* promised at the end of section 2.

Consider the neural algebras $\mathcal{N}$ on which we discussed solution algorithms in sections 2 and 3. They consist of persistent brain functions, and the solution-algorithms produced new persistent objects that extended the algebras. If we now extend such $\mathcal{N}$ to $\mathcal{N}_{rec}$ we could use that structure for coding each of the persistent objects by a natural number, using its modulus. In this coding the algebraic operations of $\mathcal{N}_{rec}$ are realised as recursive functions, thus $\mathcal{N}$ is a so-called computable algebra. By the Church-Turing Thesis any algorithm on $\mathcal{N}$ is therefore, by coding, performed as a partial recursive function, and therefore represented by a sustained element of $\mathcal{N}_{rec}$, which is what we claimed.

THE NEURAL MODEL-MATHEMATICIAN $\mathcal{N}_{rec}$

By the measure of faculties the model mathematician is the most developed (relatively speaking, to be sure). However, there is an *objection*: In order to construct $\mathcal{N}_{rec} = \langle \mathcal{B}_{rec}, \cup, \cdot, \mathbf{s}, \mathbf{p}, \mathbf{z}, \mathbf{\Gamma}, \mathbf{\Delta}, \ldots \rangle$ we used an infinite graph A_{rec}; the central objects of $\mathcal{N}_{rec}$ are conceptual rather than persistent, require an infinite neural net and have infinite modules. Let us look into this.

On infinite brains:

$\mathcal{N}_{rec}$ is, mathematically speaking, a perfectly reasonable attempt to model the brain-functions of a recursionist. But, surely, nobody would imagine that a human mathematician would infinitely think of all numbers **n**, of addition **add** and of a primality test **prime**. Human brains are in fact finite and would therefore not represent infinite conceptual objects, not completely. How does it manage? Let us put it in this way:

$\mathcal{N}_{rec}$ is an infinitistic construction of mathematics and by nature *platonic*; a finite brain $\mathcal{N}$ can only perceive conceptual objects as something like *shadows of ideas*. It realises only finite parts thereof, for example only an initial part of the number sequence. Still, such a neural model of the mathematician could be quite adequate (as far as it reaches.) Its arithmetical abilities depend of course on the implementation of the arithmetical concepts, and this could make the difference between an *idiot savant* and this *savant idiot* who has just proposed a specific $\mathcal{N}_{rec}$ as a "neural computer", which is of course far from what neuroscience has described.[10]

On a formalist standpoint:

A different reply to the objection to infinite brains would be to reflect the formalisation of mathematics in the neural algebra: the objects of mathematics are realised as sets of cascades that correspond to formal expressions of the language, axioms and proof procedures of Peano arithmetic. Such a project would yield a somewhat more pleasing model of a mathematician than that of a mere computer.

6. Coda

As a result of our thought experiment, I propose,[11] taking neural algebras as "brains", that persistent brain functions be identified with "thoughts" and that we call "conceptual thoughts", or "ideas", the sustained brain functions already provisional named conceptual. It then becomes natural to call their composition "thinking". – Have I thereby resolved the question of the title "How does the brain think?"

[10] For example: S. Dehaene, The Number Sense, Oxford U.P., 2011, and later papers, with quite different tools and goals.

[11] Switching to the, grammatical, first person instead of the "we" of an author's *pluralis modestatis*.

You could argue that by choosing the neuro-algebraic approach I have done what Goethe perceived as the typical misdirection of the mathematician: "Mathematicians are a sort of Frenchmen, when you talk to them, they translate it into their language, and then it is immediately something quite different."

This is a small beginning. It opens the view upon further potentialities for modelling faculties of the human mind in neural algebras: How does the mind arrive at problems and initiate problem solving? And how does the mind arrive at defining the focusing brain functions that call to attention and test the result of problem solving and learning? In short, what moves the mind?

Reflecting on the *Matière à pensée*,[12] more competent people, with extensive scientific background and experience in anatomical, experimental and theoretical neuroscience[13] and highly respected mathematicians[14] have written beautiful and sometimes very personal books about their experience with this *matière* and the views that they developed.

A major theme in these books is the challenge to understand *consciousness*. What can neural algebra contribute? There is one aspect of consciousness, namely self-reflection, that offers itself. *In nuce*: Suppose there is a thought "myself", represented in my neural algebra as an object M. It would satisfy the reflection equation $M \cdot M = M$. Would that be it? I doubt. Self-reflection is a much deeper subject, alive in both the sciences and the arts.[15] To develop this point, one might consider starting with an initial condition M_0 for the reflection equation, the momentary content of my attention. M_0 could simply be proprioception of my body. It could also, for example, describe my perception of the situation on my chess board, or, more interestingly, my thought, "himself", of the thoughts of my adversary. In any case, the connectome of the solution M, "myself", would be a complex set of nested cycles and I would indeed be a "strange loop". Clearly, we are still far from understanding these issues.

Apology and Thanks: I've followed relevant publications in some of the main journals for quite some time, and I'm impressed, in fact humbled, by the quality and the enormous quantity of the work on neuroscience. This pervasive influence is gratefully acknowledged. I fed it to my global neural workspace, which occasionally reassured me of being on the right track, I hope. But I beg to accept my apologies for not providing citations.

I owe much to my old friends Dana Scott and Henk Barendregt, who, when visiting me at ETH for a term in the early 1970's, introduced me to lambda calculus and combinatory logic of which neural algebra is a cousin. Cordial thanks are also due to another old friend, colleague and neighbour Klaus Hepp, who re-animated my interest in the brain and neuroscience and kept me conversant on the subject.

Dedication

This is to remember Maurice Nivat, whose passing served as a *memento mori, tempus fugit,* and made me put together these pages in his honor.

I first met Maurice in 1963 when I interviewed him for a position at the Zurich IBM Laboratory. He was a young French intellectual, with excellent background. He worked on semigroups, formal languages and automata theory which was the theoretical computer science of that time. We were trying to build up a mathematics group. The founding director of the lab, Ambros Speiser, had already hired my friend and apartmenthouse neighbour Alex Müller and my fellow student Heini Rohner with whom I used to do our physics homework. Both went on to receive Nobel prizes. Speiser started my interest in brain models, we wrote a joint contribution.[16] Forty years later, I returned to the subject, now in a much changed scientific landscape.

Maurice decided to return to Paris and to working with Schützenberger and to an influential and impressive career.

[12] Title of a delightful book on our subject, by Jean-Perre Changeux and Alain Connes: Paris, O. Jacob 1989.

[13] E.g. Koch, Christoph: Consciousness, confessions of a romantic reductionist. MIT Press, 2012; Dehaene, Stanislaw: Consciousness and the Brain. Penguin, N.Y. 2014.

[14] Such as Penrose, Roger: Shadows of the Mind. Oxford U.P. 1994.

[15] Beautifully developed by Douglas Hofstadter in his books: "Gödel, Escher, Bach and eternal golden braid", Basic Books, N.Y. 1976, and "I'm a Strange Loop", Basic Books N.Y. 2007.

[16] Engeler, E., Speiser, A.P., Zur Analogie zwischen einer elektronischen Rechenmaschine und dem Gehirn. Vierteljahresschr. d. Naturforschenden Gesellschaft in Zürich 109 (1964) 81–84.