ON THE RELEVANCE OF THE NEUROBIOLOGICAL ANALOGUE OF THE FINITE STATE ARCHITECTURE

Karl Magnus Petersson

Cognitive Neurology and Memory Research Group
F. C. Donders Centre for Cognitive Neuroimaging
University of Nijmegen, The Netherlands

Cognitive Neurophysiology Research Group R2-01
Department of Clinical Neuroscience, Karolinska Institutet
Karolinska Hospital, S-171 76 Stockholm, Sweden

Correspondance:
Karl Magnus Petersson
F. C. Donders Centre for Cognitive Neuroimaging
P.O. Box 9101, NL-6500 HB Nijmegen, The Netherlands
Email: karl.magnus.petersson@fcdonders.kun.nl

Abstract

In the present paper, we present two simple arguments for the potential relevance of the neurobiological analogue of the finite state architecture. One within the classical cognitive framework, based on the assumption that the brain is finite with respect to its memory organization, and the second within non-classical framework, based on the assumption that the brain sustains some level of 'realistic' noise and/or does not utilize infinite precision processing. These assumptions seem necessary requirements for physically realizable information processing systems. In addition we briefly review the classical cognitive framework based on Church-Turing computability and some recent non-classical approaches based on analog information processing in dynamical systems. We frame this discussion broadly in the context of recent suggestions that neurobiological and functional brain architectural constraints may have important implications for the processing architecture of the brain.

Keywords: Cognition; Computability; Complexity; Dynamical systems.

1. Introduction

In the present paper we will provide two arguments for the neurobiological relevance of the finite state architecture (FSA), or rather, its neurobiological analogue. The first will be outlined within the framework classical cognitive science while the second is outlined within a non-classical dynamical systems framework. As we proceed, we will indicate some implications for natural language representation in the human brain and (artificial) grammar learning experiments. It has recently been suggested that the task of learning an artificial grammar (AG) is a relevant model for aspects of language learning in infants (Gomez & Gerken, 2000), exploring species differences in learning (Hauser, Chomsky, & Fitch, 2002), and second language learning in adults (Friederici, Steinhauer, & Pfeifer, 2002). Recent FMRI studies indicate that language related brain regions are engaged in artificial grammar processing including Broca's region (e.g., Petersson, Forkstam, & Ingvar, in press). The cognitive domain of natural language has been shown to be relatively amenable to analysis within the classical framework of cognitive science, which suggests that isomorphic models of cognition be found within the framework of Church-Turing computability (cf., Davis, Sigal, & Weyuker, 1994). A complementary perspective is offered by neural network models of language learning and processing (for a review, see e.g., Christiansen & Chater, 2001). The network perspective can be view as a special case of the recently revived dynamical systems perspective on ('analog') information processing as a model for cognition (for overviews see e.g., Horgan & Tienson, 1996; Siegelmann, 1999; Siegelmann & Fishman, 1998). Important aspects of cognition can be equated with information processing in certain types of systems. A system will be considered an information processing system when a subclass of its states can be viewed as representational (in the sense of Jackendoff, 2002) and transitions between these can be conceptualized as implementing well-defined operations on (i.e., processing of) the representational structures. However, important constraints for models of cognition are imposed by the feasibility of realizability. These constraints can be elaborated in terms of tractable computability (Horgan & Tienson, 1996) stipulating the implementational feasibility in terms of real-time constraints and constraints set by for example the systems memory organization.

1.1 Formal languages in brief

From an extensional point of view, a formal language can be viewed as a set of strings, the set of well-formed strings belonging to the language, an E-language. The central component in the definition of an E-language is the finite alphabet of symbols or lexicon, $V = \{t_1, t_2,..., t_N\}$, over which the language is defined. The set of all possible finite symbol strings that can be generated from the alphabet V is given by Kleene-star operator $V^* = \{Ø, t_1, t_2,..., t_N, t_1t_1, t_1t_2, t_1t_3 ,..., t_{k1}t_{k2}...t_{km} ...\}$. An E-language L over V is then defined as a subset $L \subseteq V^*$; and a symbol string $s = t_{k1}t_{k2}...t_{km}$ is well-formed (or grammatical) if and only if s belongs to L, $s \in L$. The extensional definition is of limited interest from a cognitive point of view and a more fruitful generative approach (Chomsky, 1995) implies the specification of generative machinery capable of generating the language in question, an I-language, by specifying principles of combinations (e.g., Merge, Chomsky, 1995), string operations (e.g., Move, Chomsky, 1995), and non-terminal symbols over which these mechanisms operate. The generating mechanism then serves as the intentional definition of the language in the sense that a string of terminal symbols s is grammatical ($s \in L$) if and only if the formal mechanism can generate it. The class of I-languages is a subset of the class of E-languages. A straight forward cardinality argument shows that the extensional definition entails an uncountable infinity of different E-languages. However, it can be shown that only a countable infinity of different I-languages are possible, no matter how powerful methods used, as long as these methods are restricted to finitely specified representational schemes (cf. e.g., Davis et al., 1994). Informally, this means that most E-languages lack structural regularities to such a degree that they can not be completely generated by finite means. Conversely, the I-languages that can be generated by computational means, thus display a certain minimum level of structural regularity making this possible. Classes of generating mechanism can be ranked in terms of their expressivity, that is, how structurally rich the class of languages generated is. An example of this is the Chomsky hierarchy of phrase structure grammars: right-linear $\subset$ context-free $\subset$ context-sensitive $\subset$ general production grammars (cf. e.g., Davis et al., 1994), where $\subset$ denotes strict inclusion. The different levels of expressivity in the Chomsky hierarchy correspond exactly to a hierarchy of computational architectures: the finite-state, the non-deterministic push-down, the non-deterministic linearly bounded, and the (non-deterministic) Turing architecture, respectively. These computational architectures are all finitely specified with respect to their computational mechanisms and the Church-Turing hypothesis suggests that no class of finitely specified computational machines is more powerful than the Turing architecture. It turns out to be the case that the latter class can be simulated on a single universal Turing machine.

1.2 The complexity of computational mechanisms

Given that the levels in the Chomsky hierarchy are strictly inclusive it is commonly held that the class of finite-state machines represents a too restrictive to capture the syntactic phenomena found in natural languages but are commonly used in AG learning experiments. However, the expressivity of a given computational architecture depends fundamentally on both the complexity of the computational mechanism(s) and its memory organization. It will be important in the following to distinguish between the complexity of a computational mechanism [machine complexity] and the 'complexity' of its memory organization. However, we will first briefly outline one formulation of the computational framework of classical cognitive science. We will consider the conceptually simpler case of a deterministic ('cognitive') transition function. This is no restriction since non-deterministic transition relations do not add any computational power. Let Σ be the input space (input alphabet) of inputs σ ($\sigma \in \Sigma$), Ω the state space of internal states ω ($\omega \in \Omega$), and Λ the output space of outputs λ ($\lambda \in \Lambda$). The possible transitions T between internal states are then determined by a function T: $\Omega x \Sigma \to \Omega$ and the possible outputs are determined by a function R: $\Omega x \Sigma \to \Lambda$. In other words, suppose at processing step n, the system receives input

$\sigma(n)$ when in state $\omega(n+1)$, then the system changes state into $\sigma(n)$, while generating the output $\lambda(n+1)$ according to:

$$\omega(n+1) = T[\omega(n), \sigma(n)] \qquad (1)$$

$$\lambda(n+1) = R[\omega(n), \sigma(n)] \qquad (2)$$

In this way, the processing system traces a trajectory in state space, …, $\omega(n)$, $\omega(n+1)$, …, while reading the input stream …, $\sigma(n)$, $\sigma(n+1)$, …, and generating the output sequence …, $\lambda(n)$, $\lambda(n+1)$, …. Within the framework of Church-Turing computability it is assumed that Σ, Ω, and Λ are all finite and that T and R are finitely specified. Note that we have here taken a dynamical systems perspective on computability. What has not been explicitly described here is the system's memory organization, cf., Table 1:

Architecture	**Complexity**	**Memory organization**			
		Iternal states	***Registers***	***Stack***	***Accessability***
FSA	Finite	Finite			
PDA	Finite	Finite		Unlimited	Top of stack
LBA	Finite	Finite	Unlimited[1]		Random access
URA	Finite	Finite	Unlimited		Random access

[1] Linearly bound in the input size with a universal constant.

Tabel 1. The Chomsky hierarchy and the memory organization of respective architecture. FSA = finite state architecture, PDA = push-down architecture, LBA = lineraly bounded architecture, URA = unlimited register architecture (equivalent to the Turing architecture).

In the following we will focus on just one aspect of memory organization, that is, whether the storage capacity is finite or infinite. This aspect is crucial for language expressivity. Two aspects of expressivity are of importance in this context: the number of sentences that a generative mechanism can express (finite or infinite) and the different types of recursive structure that can be expressed in these sentences. In all computational architectures, including those corresponding to the classes of grammars in the Chomsky hierarchy, the computational mechanism is finitely specified and the transition function T: $\Omega x \Sigma \rightarrow \Omega$ can always be realized in a finite state architecture. In other words, respect to the computational mechanism subserving 'cognitive' transitions between internal states there is no distinction in terms of machine complexity between the different classes of computational machines in the Chomsky hierarchy, they are all finite in this respect (cf., Savage, 1998). However, as indicated by the strict inclusion in the Chomsky hierarchy, there are differences in language expressivity. The strict inclusion springs from an inseparable interaction between the finitely specified generating mechanism(s) and difference in memory organization. A crucial determinant of structural expressivity is the access to memory with no restrictions (i.e., infinite or potentially infinite storage capacity), with which the computational machinery can interact. In a fundamental sense, it is the characteristics of the memory organization of the different computational architectures, which allow them to re-use their processing capacities recursively to realize functions of high complexity or generate structurally rich languages. From a neurophysiological perspective it seems reasonable to assume that the human brain possesses finite memory resources, both with respect to long-term memory and, perhaps more relevant here, short-term working memory. If this is the case, one important implication form the point of view of classical cognitive science is that cognitive brain functions can be formulated as a finite state architecture.

2. Learnability

Chomsky (1986) suggested that not only is prior innate structures or constraints necessary, but that is reasonable to assume that these prior constraints represents linguistically specific competence in the form of a specific initial state of the faculty of language and a specific language acquisition device. This conclusion is often reinforced by referring to an early formal learning theoretical result of Gold (1967), which basically states that under some circumstances no super-finite class of languages, including the class of regular languages, is learnable from positive examples alone (see also e.g., Jain, Osherson, Royer, & Sharma, 1999). It has also been suggested that this is the case when statistical learning mechanisms are employed (Nowak, Komarova, & Niyogi, 2002). It should be noted that the assumption of prior innate constraints just transfer the problem of learnability to another domain, presumably that of phylogenetic development and a problem of the evolutionary origin of language. Furthermore, already Gold (1967) noted that under suitable other circumstances this (un)learnability paradox can be avoided. This may for example include the existence and effective use of explicit negative feedback, prior restrictions on the class of possible languages, or if there are prior restrictions on the possible language experiences that can occur, that is, prior restrictions on the characteristics of the possible language environments. Recent results in formal learning theory confirm Gold's (1967) suggestion that, if the class of possible languages is restricted, then it is possible to learn infinite languages in infinite classes of formal languages from positive examples (Shinohara, 1994). It should be noted that these constraints are of a general kind and not necessarily 'linguistically specific'. There exists classes of formal languages rich enough to encompass the string sets of human languages and at the same time being identifiable from a finite sequence of positive examples (Scholz & Pullum, 2002). Furthermore, the acquisition task becomes potentially more tractable if there are additional structure in the input or if only probable approximate identification success is required. One possibility is to generate expectations or predictions based on an internal model. If the learning system has access to or can acquire a forward model, this can be used for model dependent prediction. This entails the possibility of an unsupervised learning framework in which the difference between input and predictions drives the learning process. A simple example of this is the predictive simple recurrent neural network (SRN) architecture (e.g., Elman, 1990). Recent connectionist modeling suggest that this may be a viable approach to finite recursion (Christiansen & Chater, 1999).

3. Analog information processing in dynamical systems

The different architectures of the Chomsky hierarchy allows for different types of recursion, a feature thought to be at the core of the language faculty (Hauser et al., 2002; Jackendoff, 2002). For example, the finite state architecture support unlimited concatenation recursion and can support limited (i.e., finite) recursion of general type. This is also characteristic of human language performance (cf. e.g., Christiansen & Chater, 1999). Unlimited embedding recursion is supported by unlimited push-down architecture, while unlimited cross-dependency recursion requires an unlimited linearly-bound architecture. None of these latter capacities appears to be characteristic of human performance. These observations may explain the sometimes surprising performance of simple processing architectures that are finitely specified both in terms of computational machinery and memory organization, that is, physically realizable processing systems. For example, a recent study, using the SRN architecture (Elman, 1990), attempted to model certain aspects of human language processing with respect to different types of finite recursion (Christiansen & Chater, 1999). The SRN architecture can be viewed as a simple network analogue of the finite state architecture. More specifically, at time point n+1, the output $\lambda(n+1)$ of the SRN is a function of the input $\sigma(n)$ and the previous internal state $\omega(n)$, while the new internal state $\omega(n+1)$ (i.e., the state of the hidden layer of computational units) is stored (or copied) into an additional short-term memory layer. The new internal state $\omega(n+1)$ is a function of the previous internal state $\omega(n)$ and the input $\sigma(n)$, that is:

$$\lambda(n+1) = N[\omega(n),\ \sigma(n)] \qquad (3)$$

$$\omega(n+1) = C[\omega(n),\ \sigma(n)] \qquad (4)$$

It is clear that the form of equations (3) and (4) are identical to (2) and (1), respectively. Already, McCulloch and Pitts (1943) showed that that the class of networks of so-called McCulloch-Pitts neurons (simple threshold units) is equivalent to the class of finite state machines (cf., Kleene, 1956). Information processing neural networks can be viewed as resulting from parallel interactions in a dynamical system and learning, in for example an SRN, is realized by making the function N dependent on learning parameters α. In other words, instead of a single function N, we are assuming a model space M = {N[α]| where α is accessible or can be instantiated by the processing system}. Learning, or development of the system, can then be conceptualized as a trajectory in its accessible model space M driven by the interaction with the environment (possibly in conjunction with innately specified developmental processes). Thus as the system develops, it traces out a trajectory in M determined by its adaptive dynamics L:

$$\alpha(n+1) = L(\alpha(n), \sigma(n), n) \quad (5)$$

given some initial state $\alpha = \alpha_0$. Here we have made L explicitly dependent on time, n, in order to capture the idea of innately specified developmental processes (e.g., 'maturation') as well as the possible dependence on the previous developmental history, thus slightly generalizing the standard SRN framework. Thus, in a general sense, information processing and learning can be viewed as coupled dynamical system, where the processing dynamics (e.g., eq. (4)) is coupled with the learning dynamics (e.g., eq. (5)).

The SRN framework briefly outlined differs in information processing capacity from the finite state architecture to the extent it can be viewed as an 'analog' processing model. More, specifically, to the extent that the framework takes advantage of the continuum property of real numbers (or, possibly any dense sub-set, e.g., the rational numbers) in conjunction with infinite processing precision. In the context of computer simulation of SRNs, one may note that computation with finite precision numbers effectively is a FSM simulation. However, from an idealized mathematical point of view, the SRN architecture can be viewed as an 'analog' information processing system, a simple network analogue of the finite state architecture (cf. e.g., Casey, 1996; Siegelmann, 1999). Thus, the 'analog' network approach offers an interesting possibility to model cognition within a non-classical framework. One advantage of a network approach is that memory characteristic arise naturally from the particular network architecture chosen, that is, are inherent in the architecture, instead of having to be imposed in terms of for example an arbitrary finite storage capacity. In general, analog recurrent neural networks can be viewed as a finite number of analog registers (e.g., the 'membrane potential' of neurons) exchanging and processing information through their transfer functions and network interconnectivity.

Several non-standard models of computation (i.e., information processing) have recently been outlined (for reviews see e.g., Arbib, 2003; Siegelmann, 1999; Siegelmann & Fishman, 1998). For example, one way to generalize the Church-Turing framework of computability is to employ analog instead of discrete representations (Siegelmann, 1999). Another is to move from static to adaptive processing, that is, to use parameterized models in combination with adaptive dynamics, as exemplified above. Note that the universal Turing machine can be viewed as a parameterized architecture incorporating all possible Turing machines (i.e., parameterized by the 'program number' in a von Neumann type architecture, cf., Davis et al. (1994)). In an early study, Pollack (1987) showed that certain types of connectionist networks can simulate Turing machines. In the case of recurrent networks, the processing power depends on, among other things, the type of numbers utilized as adaptive weights, including natural-, rational-, and real numbers, corresponding to networks that are computationally equivalent to finite state machines, Turing machines, and non-uniform processing models, respectively (for a recent review see Siegelmann, 1999). Furthermore, large classes of dynamical systems do not have greater processing capacities than the analog recurrent network architecture (Siegelmann, 1999). That these capacities for information processing can be captured within the framework of ('analog') continuous dynamical systems are less surprising given the enormous richness of continuous dynamical systems (Lasota & Mackey, 1994). What is perhaps more surprising is that an important extent of this richness is possible to capture in simple recurrent neural network architectures. However, the type of analog information processing which takes full advantage of and utilizes infinite precision in processing is a mathematical idealization and it should be noted that there appears to be several brain internal noise sources (Gerstner & Kistler, 2002; Koch, 1999), even though the nature of these are not fully understood (cf., Gerstner & Kistler, 2002). Though the analog processing principles based on attractor dynamics in recurrent networks may show a degree of noise tolerance and architectural imprecision (Siegelmann, 1999), because of the dependence on infinite precision, the capacities of these networks are sensitive to system noise, preventing the networks from recognizing languages generated by arbitrary FSA in the context of 'realistic' noise sources (Maass & Sontag, 1999). Within the frame work of analog dynamical systems, it is clear that any reasonable model of the brain (or any sub-system) will have a compact manifold as its state space in some finite dimensional real space (cf., Lasota & Mackey, 1994). Here compactness represents the natural generalization of finite state spaces used in the classical computational framework. Furthermore, finite precision processing or a non-zero 'realistic' noise level would have the effect of coarse graining the state space, effectively discretize it into a finite number of 'voxels' of indistinguishable states (due to compactness). It thus appears that, in such a situation, the analog dynamical systems would behave, to some degree of approximation, as finite state analogue.

4. Conclusion

It has been suggested that classical cognitive models of language can be viewed as approximate abstract higher level descriptions of the properties of the brain networks. It is interesting to note that neurobiological and functional brain architectural constraints may have important implications for the characteristics of the language faculty, as suggested by Hauser, Chomsky, and Fitch (2002) and Jackendoff (2002). This offers the possibility that neural network approaches in conjunction with learning theory may represent an important opportunity for a deeper understanding of the connection between the human brain and the faculty of language. Neural networks represent a class of finitely parameterized models with universal approximation properties, in addition to being neurobiologically in flavour, and in this sense closer to neural processing. In the present paper, we have offered two simple arguments for the potential relevance of the neurobiological analogue of the finite state architecture. One classical, based on the assumption of the brain is finite with respect to its memory organization, and the second non-classical, based on the assumption that the brain sustains some level of 'realistic' noise and/or does not utilize infinite precision processing. These assumptions seem necessary requirements for physically realizable information processing systems.

References

Arbib, M. A. (Ed.). (2003). *The Handbook of Brain Theory and Neural Networks* (2 ed.). Cambridge, MA: MIT Press.

Casey, M. (1996). The dynamics of discrete-time computation with application to recurrent neural networks and finite state machine extraction. *Neural Computation, 8*, 1135-1178.

Chomsky, N. (1986). *Knowledge of Language*. New York: Praeger.

Chomsky, N. (1995). *The Minimalist Program*. Cambridge, MA: MIT Press.

Christiansen, M. H., & Chater, N. (1999). Toward a connectionist model of recursion in human linguistic performance. *Cogn. Sci., 23*, 157-205.

Christiansen, M. H., & Chater, N. (Eds.). (2001). *Connectionist Psycholinguistics*. Norwood, NJ: Ablex Publishing.

Davis, M. D., Sigal, R., & Weyuker, E. J. (1994). *Computability, Complexity, and Languages: Fundamentals of Theoretical Computer Science* (2 ed.). San Diego, CA: Academic Press.

Elman, J. L. (1990). Finding structure in time. *Cogn. Sci., 14*, 179-211.

Friederici, A. D., Steinhauer, K., & Pfeifer, E. (2002). Brain signatures of artificial language processing. *Proc. Natl. Acad. Sci. USA, 99*, 529-534.

Gerstner, W., & Kistler, W. (2002). *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge, UK: Cambridge University Press.

Gold, E. M. (1967). Language identification in the limit. *Information and Control, 10*, 447-474.

Gomez, R. L., & Gerken, L. (2000). Infant artificial language learning and language acquisition. *TICS, 4*, 178-186.

Hauser, M. D., Chomsky, N., & Fitch, W. T. (2002). The faculty of language: what is it, who has it, and how did it evolve? *Science, 298*(5598), 1569-1579.

Horgan, T., & Tienson, J. (1996). *Connectionism and the Philosophy of Psychology*. Cambridge, MA: MIT Press.

Jackendoff, R. (2002). *Foundations of Language: Brain, Meaning, Grammar, Evolution*. Oxford, UK: Oxford University Press.

Jain, S., Osherson, D., Royer, J. S., & Sharma, A. (1999). *Systems That Learn*. Cambridge, MA: MIT Press.

Kleene, S. C. (1956). Representation of events in nerve nets and finite automata. In C. E. Shannon & J. McCarthy (Eds.), *Automata Studies* (pp. 3-42). Princeton, NJ: Princeton University Press.

Koch, C. (1999). *Biophysics of Computation: Information Processing in Single Neurons*. New York: Oxford University Press.

Lasota, A., & Mackey, M. C. (1994). *Chaos, Fractals, and Noise: Stochastic Aspects of Dynamics*. New York: Springer-Verlag.

Maass, W., & Sontag, E. D. (1999). Analog neural nets with Gaussian or other common noise distributions cannot recognize arbitrary regular languages. *Neur. Comp., 11*, 771-782.

McCulloch, W. S., & Pitts, E. (1943). A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophysics*(5), 115-133.

Nowak, M. A., Komarova, N. L., & Niyogi, P. (2002). Computational and evolutionary aspects of language. *Nature, 417*, 611-617.

Petersson, K. M., Forkstam, C., & Ingvar, M. (in press). Artificial syntactic violations activates Broca's region. *Cognitive Science*.

Pollack, J. B. (1987). *On Connectionist Models of Natural Language Processing.* Unpublished PhD, University of Illinois, Urbana.

Savage, J. E. (1998). *Models of Computation: Exploring the Power of Computing*. Reading, MA: Addison-Wesley.

Scholz, B. C., & Pullum, G. K. (2002). Searching for arguments to support linguistic nativism. *Ling. Rev., 19*, 185-223.

Shinohara, T. (1994). Rich classes inferable from positive data: Length-bounded elementary formal systems. *Information and Computation, 108*, 175-186.

Siegelmann, H. T. (1999). *Neural Networks and Analog Computation: Beyond the Turing Limit*. Basel, Switzerland: Birkhäuser.

Siegelmann, H. T., & Fishman, S. (1998). Analog computation with dynamical systems. *Physica D, 120*, 214-235.