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

Karl Magnus Petersson

F. C. Donders Centre for Cognitive Neuroimaging, University of Nijmegen, The Netherlands

Cognitive Neurophysiology Research Group, Karolinska Institutet, Stockholm, Sweden

Centre for Intelligent Systems, Universidade do Algarve, Portugal

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

We present two simple arguments for the potential relevance of a neurobiological analogue of the finite state architecture. The first is based on the assumption that the brain is finite with respect to its memory organization, and the second is based on the assumption that the brain sustains some level of noise and/or does not utilize infinite precision processing. We briefly review the classical cognitive framework based on Church-Turing computability and non-classical approaches based on analog processing in dynamical systems.

Keywords: Cognition; Computability; Complexity; Dynamical systems.

1. Introduction

In the present paper we will provide two arguments for the neurobiological relevance of the generic finite state architecture (FSA), or rather, its approximate neurobiological analogue. The first will be outlined within the framework classical cognitive science and the second within a general dynamical systems framework. We will also indicate some implications for natural language processing and artificial grammar learning. It has recently been suggested that the task of learning an artificial grammar is a relevant model for aspects of language learning in infants [1], exploring species differences in learning [2], and second language learning in adults [3]. Recent FMRI studies also indicate that language related brain regions are engaged in artificial grammar processing [4]. Some aspects of natural language are amenable to an analysis within the classical framework of cognitive science. This framework suggests that isomorphic models of cognition can be found within the framework of Church-Turing computability [cf., 5]. A complementary perspective is offered by network models of language processing [for a review, see e.g., 6]. The network perspective represents a special case of the recently revived dynamical systems perspective on analog information processing as a model for cognition. Here a system is considered an information processing system when a subclass of its states can be viewed as cognitive [in the sense of 7, pp. 19-23] and transitions between these can be conceptualized as processing of these cognitive structures. Important constraints on cognitive models are imposed by realizability requirements. These constraints can be elaborated in terms of tractable computability, including constraints on real-time processing and memory organization.

1.1 Formal languages

From an extensional point of view, a formal language can be viewed as a set of strings, an E-language. Given a the finite alphabet of terminal symbols V over which the E-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* [cf., 5]. An E-language L over V is then defined as a subset $L \subseteq V^*$; and a string s is well-formed if and only if it belongs to L ($s \in L$). This extensional definition is of limited interest from a cognitive point of view and a more fruitful generative approach [8] entails the specification of machinery capable of generating the language in question, an I-language, by specifying principles of combinations [e.g., Merge, cf., 8], string operations [e.g., Move, cf., 8], and non-terminal symbols over which these mechanisms operate. The generative machinery serves as an intentional definition of the language in the sense that a string of terminal symbols is well-formed 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. It can also be shown that only a countable infinity of different I-languages are possible, no matter how powerful generative methods used, as long as these are restricted to finitely specified representational schemes [cf. e.g., 5]. 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 display a certain minimum level of structural regularity. Classes of generating mechanism can be ranked in terms of their expressivity, that is, how structurally rich the generated languages are. An example of this is the Chomsky hierarchy of phrase structure grammars: right-linear $\subset$ context-free $\subset$ context-sensitive $\subset$ general production grammars, where $\subset$ denotes strict inclusion [cf. e.g., 5]. 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, and the latter class can be simulated on a single universal Turing machine [cf. e.g., 5].

1.2 The complexity of computational mechanisms

Given that the levels in the Chomsky hierarchy are strictly inclusive it is commonly held that the FSA is too restrictive to capture all syntactic phenomena found in natural languages. The computational expressivity of any given architecture depends fundamentally on both the complexity of the computational mechanism(s) and its memory organization. The computational framework of classical cognitive science can be formulated from a dynamical systems point of view. We will consider the conceptually simpler case of a deterministic transition function. This is no restriction since non-deterministic transition relations do not add computational power [cf. e.g., 5]. Let Σ be the input space ($\sigma \in \Sigma$), Ω the state-space of internal states ($\omega \in \Omega$), and Λ the output space ($\lambda \in \Lambda$). The possible transitions T between internal states are then determined by a function T: $\Omega x\Sigma \rightarrow \Omega$ and the outputs are determined by a function R: $\Omega x\Sigma \rightarrow \Lambda$. In other words, at processing step n, the system receives input $\sigma(n)$ when in state $\omega(n)$, then the system changes state into $\omega(n+1)$, 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 system traces a trajectory in state-space, …, $\omega(n)$, $\omega(n+1)$, …, while receiving the input …, $\sigma(n)$, $\sigma(n+1)$, …, and generating an output sequence …, $\lambda(n)$, $\lambda(n+1)$, …. Within the framework of Church-Turing computability Σ, Ω, and Λ are all finite and thus T and R are finitely specified. Here we have not explicitly described the system's memory organization (cf. Table 1). 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. We will focus on just one aspect of memory organization, the storage capacity; in particular, whether this is finite or infinite. This turns out to be crucial for the expressivity of the system. One important aspect of expressivity is the types of recursive structure that are expressed in these strings.

[Table 1]

In all classical architectures, the transition function T: $\Omega x\Sigma \rightarrow \Omega$ can be realized in a FSA. Thus, with respect to the mechanism subserving transitions between internal states there is no fundamental distinction in terms of machine complexity between the different computational architectures [Table 1, cf., 9]. However, as indicated by the strict inclusion in the Chomsky hierarchy, there are differences in expressivity. This is fundamentally related to the interaction between the generating mechanism and the available memory organization. The most important determinant of structural expressivity is the availability (or not) of infinite storage capacity. Thus, it is the characteristics of the memory organization, which in a fundamental sense, allow the architecture to recursively use its processing capacities inherent in T, to realize functions of high complexity or achieve high levels of expressivity. From a neurophysiological perspective it seems reasonable to assume that the brain only possesses finite memory resources, both with respect to short-term and long-term memory. If this is the case, assuming the classical cognitive science framework, one implication of this discussion is that brain functions can be formulated within the FSA [in the level 1 sense of 10].

2. Learnability

Chomsky [11] suggested that prior innate constraints are necessary in order to acquire a natural language. More specifically, he suggested that these constraints represent a linguistically specific competence in the form of a specific language acquisition device and a specific initial state of the language faculty. It should be noted that this assumption transfers the problem of learnability to a problem of the evolutionary origins of language. Moreover, this suggestion is often reinforced by invoking an early result of Gold [12], which states that under some circumstances no super-finite class of languages is learnable from positive examples alone. It has also been suggested that this is the case when statistical learning mechanisms are employed [13]. Gold [12], however, noted that the learnability problem can be avoided under suitable assumptions, including the existence and effective use of explicit negative feedback, prior restrictions on possible languages, or restrictions on possible language environments. Recent results in learning theory confirm this. For example, under general constraints (i.e., not linguistically specific) it is possible to learn infinitely rich classes of infinite languages from positive examples [14]. Furthermore, the acquisition task becomes potentially more tractable if there are additional structure in the input or if only approximate identification is required. One possibility is to generate expectations based on an internal model for prediction. Within an unsupervised learning framework, the internal model can be created through a learning process (e.g., self-organized based on general constraints) which is driven by the difference between input and internally generated predictions. A simple example of this is the predictive simple recurrent network (SRN) architecture [e.g., 15]. Recent results suggest that this may be a viable approach to finite recursion [6].

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 [2, 7]. For example, the FSA supports unlimited concatenation recursion and can support finite recursion of general type, which is also characteristic of human performance. Unlimited embedding recursion is supported by the push-down architecture, while unlimited cross-dependency recursion requires an linearly-bound architecture [cf., Table 1 and 5]. None of these are human characteristics. These observations explain the capacity of simple architectures to model human performance; for example, a recent study used discrete-time SRNs to model different types of finite recursion [6]. The discrete-time SRN can be viewed as a simple network analogue of the FSA. 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 copied to short-term memory layer:

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

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

It is clear that the form of (3) and (4) are identical to equations (2) and (1), respectively. An early result of McCulloch and Pitts [16] showed that the class of networks of thresholding units is equivalent to the FSA.

Generally, learning and development can be realized by making the system function, for example N of the SRN, dependent on learning parameters α, yielding a model space M = $\{N[\alpha] \| \alpha$ realizable by the system$\}$. Learning (or development) can then be conceptualized as a trajectory in the model space M, driven by environmental interaction in conjunction with constraints and system plasticity, determined by the adaptive dynamics L of the system:

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

given an initial state $\alpha = \alpha_0$; L is explicitly dependent on time in order to capture the idea of innately specified developmental processes and possible dependence on the previous developmental history. In general, 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 (eq. 5). If we replace the discrete time n with a continuous time t and specify the differential time changes, instead of the discrete-time changes, we have the general continuous-time case. The SRN differs in processing capacity from the FSA to the extent it can be viewed as an analog model; that is, to the extent that the SRN takes advantage of infinite processing precision on real numbers (or any dense subset) in state-space and/or in time. Generally, analog network approaches offer interesting possibilities to model cognition within a non-classical framework. For example, memory characteristics arise naturally from particular network architectures. An analog recurrent neural network can generally be viewed as a finite number of analog registers (e.g., the ‘‘membrane potential’’) that processes information interactively through its network topology and the transfer functions of its processing units. Several non-standard computational models have recently been outlined [for a review see e.g., 17], including generalizations of the Church-Turing framework: analog instead of discrete states and/or time in combination with infinite processing precision; parameterized models in combination with adaptive dynamics (note: the universal Turing machine can be viewed as parameterized by the "program number", see e.g., Davis et al., 1994).

It is an interesting fact that the processing power of recurrent networks depends on, among other things, the type of numbers utilized as adaptive weights (e.g., natural, rational, real numbers corresponds precisely to finite state, Turing, and super-Turing models), and a very large class of dynamical systems do not posses greater processing capacities than the analog recurrent network architecture [17]. However, the dependence on infinite precision processing implies that these capacities generally are sensitive to system noise. Importantly, there appears to be several brain internal noise sources [e.g., 18]. Now, it seems clear that any reasonable analog model of a brain system will have a state-space in the form of a compact manifold (i.e., closed and bounded) in some finite dimensional real space. Here the mathematical property of compactness represents the natural generalization of finiteness in the classical framework. Moreover, finite precision computations or realistic noise levels would have the effect of coarse graining the state-space, thus effectively discretizing the state-space into a finite number of "voxels" of indistinguishable states. This follows immediately from the compactness property. It thus appears that even if we model a brain system as an analog dynamical system, this would behave (approximately) as a finite state analogue. Under the additional assumption of finitely available processing time, the same conclusions follow in the case of continuous-time evolution of state variables if finite temporal precision is assumed.

4. Conclusion

It has been suggested that classical cognitive models of language can be viewed as approximate abstract descriptions of properties of the brain. It is well accepted that neurobiological and functional brain constraints have important implications for the characteristics of the language faculty [2, 7]. In the present paper, we have offered two simple arguments for the potential relevance of the neurobiological analogue of the finite state architecture. The first 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 of a compact state-space in conjunction some level of noise or finite precision processing. Either assumption seems necessary for physically realizability.

References

[1] Gomez, R.L. and L. Gerken, Infant artificial language learning and language acquisition. TICS, 2000. 4: 178-186.

[2] Hauser, M.D., N. Chomsky, and W.T. Fitch, The faculty of language: what is it, who has it, and how did it evolve? Science, 2002. 298: 1569-79.

[3] Friederici, A.D., K. Steinhauer, and E. Pfeifer, Brain signatures of artificial language processing. Proc. Natl. Acad. Sci. USA, 2002. 99: 529-534.

[4] Petersson, K.M., C. Forkstam, and M. Ingvar, Artificial syntactic violations activate Broca's region. Cognitive Science, 2004. 28: 383-407.

[5] Davis, M.D., R. Sigal, and E.J. Weyuker, Computability, Complexity, and Languages: Fundamentals of Theoretical Computer Science. 2nd ed. 1994, San Diego, CA: Academic Press.

[6] Christiansen, M.H. and N. Chater, eds. Connectionist Psycholinguistics. 2001, Ablex Publishing: Norwood, NJ.

[7] Jackendoff, R., Foundations of Language: Brain, Meaning, Grammar, Evolution. 2002, Oxford, UK: Oxford University Press.

[8] Chomsky, N., The Minimalist Program. 1995, Cambridge, MA: MIT Press.

[9] Savage, J.E., Models of Computation: Exploring the Power of Computing. 1998, Reading, MA: Addison-Wesley.

[10] Marr, D., Vision: A Computational Investigation into the Human Representation and Processing of Visual Information. 1982, New York: W. H. Freeman and Company.

[11] Chomsky, N., Knowledge of Language. 1986, New York: Praeger.

[12] Gold, E.M., Language identification in the limit. Information and Control, 1967. 10: 447-474.

[13] Nowak, M.A., N.L. Komarova, and P. Niyogi, Computational and evolutionary aspects of language. Nature, 2002. 417: 611-617.

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

[15] Elman, J.L., Finding structure in time. Cogn. Sci., 1990. 14: 179-211.

[16] McCulloch, W.S. and E. Pitts, A logical calculus of the ideas immanent in nervous activity. 1943. Bull. Math. Biophysics 5: 115-133.

[17] Siegelmann, H.T., Neural Networks and Analog Computation: Beyond the Turing Limit. 1999, Basel, Switzerland: Birkhäuser.

[18] Gerstner, W. and W. Kistler, Spiking Neuron Models: Single Neurons, Populations, Plasticity. 2002, Cambridge, UK: Cambridge University Press.

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

Architecture	**Complexity**	**Memory organization**			
		Internal 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.