Explanatory Aspirations and the Scandal of Cognitive Neuroscience

Ross W. GAYLER[a], Simon D. LEVY[b], and Rens BOD[c]
[a] *Philosophy Program, La Trobe University, Bundoora VIC 3086, Australia (r.gayler@gmail.com)*
[b] *Department of Computer Science, Washington and Lee University, Lexington VA 24450, USA (levys@wlu.edu)*
[c] *Institute for Logic, Language and Computation, University of Amsterdam, Plantage Muidergracht 24, Amsterdam NL-1018 TV, The Netherlands (rens.bod@uva.nl)*

Abstract. In this position paper we argue that BICA must simultaneously be compatible with the explanation of human cognition and support the human design of artificial cognitive systems. Most cognitive neuroscience models fail to provide a basis for implementation because they neglect necessary levels of functional organisation in jumping directly from physical phenomena to cognitive behaviour. Of those models that do attempt to include the intervening levels, most either fail to implement the required cognitive functionality or do not scale adequately. We argue that these problems of functionality and scaling arise because of identifying computational entities with physical resources such as neurons and synapses. This issue can be avoided by introducing appropriate virtual machines. We propose a tool stack that introduces such virtual machines and supports design of cognitive architectures by simplifying the design task through vertical modularity.

Keywords. cognitive neuroscience, Jackendoff's challenges, Neural Engineering Framework, Vector Symbolic Architecture, Data-Oriented Processing

1. On Being Biologically Inspired

Biologically Inspired Cognitive Architecture implies the choice of biological features to use as the inspiration, which is difficult because biological systems are evolved and it is not clear for any feature whether it is a functional essence or an evolutionary accident. Lacking a sound, accurate theory of biologically implemented cognition such a choice necessarily constitutes a bet by the researcher that the chosen inspiration yields a productive organising framework rather than a dead end. Many BICA researchers take inspiration from current cognitive neuroscientific explanations, which we argue have major gaps. Further, these gaps are generally unacknowledged, making it more likely that BICA researchers will be misled by them. We analyse these gaps and propose an alternative source of inspiration which we hope will be more productive.

1.1. "What I cannot create I do not understand"

The aim of cognitive neuroscience is to explain cognition from biological substrates, but what is the desired level of explanation? Richard Feynman wrote "What I cannot create, I do not understand" [1]. Taking this as the test of understanding, cognitive neuroscientists should seek the knowledge that would enable, in principle, the design of a system of neurons and synapses that implements cognitive functions. This standard is even more relevant to BICA, where the explicit aim is to build cognitive systems. But is current neuroscientific knowledge enough to build practical systems?

Cognitive neuroscience tends to be structured around levels of physical organisation in the nervous system (e.g. synapses, neurons, cortical microcircuits, and neuroanatomy). Research spanning adjacent levels of physical organisation tends to be reductionist, explaining higher level properties that are necessary consequences of the lower-level, while ignoring higher-level properties that are supported by but not necessary consequences of the lower-level. Cognitive neuroscience explanations tend to jump in a single, explanatory leap from the studied level of physical organisation to cognitive behaviours (e.g. from neural spikes to semantics [2]). Such explanations have their uses but leave gaps with respect to implementation (which is central to BICA).

Even if our understanding of a single level of organisation were perfect it would not necessarily explain cognition. Consider the challenge facing a Martian scientist given the task of understanding a web search engine sufficiently well to build one. This would require an understanding of transistors, logic gates, processors, computer clusters, operating systems, distributed file systems, indexing techniques, and so on. In this analogy an explanation of cognition in terms of synapses and neurotransmitters is equivalent to attempting to explain a web search engine only in terms of the quantum physics of semiconductors. Other neuroscience approaches do not fare any better in this analogy: neuroanatomy is the equivalent of identifying boxes and the cables between them; functional MRI is the equivalent of measuring power consumption of the boxes; psychology examines the input/output relationships and their timings. These are all useful approaches in their own way but they don't directly address how to build such a system. A credible basis for implementation must address all the appropriate levels of functional organisation in going from physical observations to cognitive behaviour,.

1.2. Problems with Credible Neural System Models – Functionality and Scaling

It is not enough for a neural model to include intervening levels of functional organisation; they must be appropriate levels. Many apparently credible neural models have problems that make their use in BICA questionable. We consider two classes of models: neurally realistic (e.g. [3]) and connectionist models that abstract away most of the neurophysiology in order to focus on the computational properties (e.g. [4]). Almost all these models lack the functionality required to properly implement cognitive behaviour. Of the models that are functionally adequate, most don’t scale up satisfactorily. The scandal of cognitive neuroscience is that these problems exist and are not generally acknowledged (which makes them more dangerous to BICA).

1.2.1. Lack of Functionality

Jackendoff observed that neural models of linguistic functions fail to deal with fundamental issues of the domain [5] and posed four challenges for cognitive neuroscience. Although he framed the problems in linguistic terms, he believed that they are problems of general cognition rather than being uniquely linguistic. The essence of the challenges is the neural instantiation of rapid construction and transformation of compositional structures. Although such structures are typically taken to be the stuff of symbolic, propositional processing the relevance of composition is more general. For example, Barsalou proposes that cognition is inherently perceptual, consisting of the execution of perceptual simulations, and notes the need to compose simulations to account for productivity [6]. Even if you believe that some parts of cognition do not require compositional structures, the fact remains that linguistic cognition must be accounted for. Jackendoff's challenges are briefly described as:

1. *"The massiveness of the binding problem"* Binding refers to structural association of representations (i.e. which goes with what, and how). Given a red square and a blue circle, binding consists of ensuring that "*red*" is associated as the colour of "*square*" and "*blue*" as the colour of "*circle*" and no other possible associations occur.

Some cognitive tasks, such as comprehending this paper, require many novel structural relations to be tracked. This is challenging because of the large number of (potentially novel) bindings and the speed with which they must be created and manipulated. These are difficult problems for neural models that dedicate specific physical resources (e.g. connections) to represent specific entities such as relations, because they imply the implausibly rapid creation or recruitment of physical resources.

2. *"The problem of 2"* This challenge refers to the need to represent and manipulate multiple instances of the same type. For example, comprehending a scene in which a red cup is beside a blue cup requires two separate representations of cups which must enter into different relations with other entities. This is difficult for models that associate meanings with specific physical resources (e.g. neurons) because separate resources are required for each represented instance of the entity. These physical resources need to be created or recruited on demand and kept distinct, while simultaneously sharing and learning any properties common across the instances.

3. *"The problem of variables"* People can recognise and manipulate an infinite variety of situations using only finite representational resources. This productivity is construed as arising from the use of variables; placeholders which may contain arbitrary values. This is a problem for neural models that represent a variable with specific physical resources because those resources must be capable of representing every value that the variable can take (including complex structures). Alternatively, the variable and every possible value could be represented by separate physical resources and the variable/value relation between them captured by binding, but this leads to the first and second challenges.

4. *"Binding in working memory vs. long-term memory"* This concerns the functional equivalence of working memory and long-term memory. A composite structure may, at different times, be instantiated in both working memory and long-term memory and these instantiations should be functionally equivalent. Typical neural models suggest a lack of equivalence because working memory representations are implemented using activation levels or temporal synchrony, whereas long-term memory is implemented using synaptic connectivity. These implementations require mechanisms to translate representations between them and allow the representations to interact. Standard connectionist models do provide such mechanisms, but the greater challenge is to ensure that both implementations have equivalent representational capabilities with respect to composite structures, binding, and variables.

1.2.2. Lack of Scaling

Most neural models of cognition fail at least one of Jackendoff's challenges and are thus functionally inadequate. Of the few models than can claim to meet the challenges, most rely on allocation of specific physical resources (e.g. neurons) to represent specific entities or relationships. The consequence of this direct correspondence between the implementing mechanisms and the representations is that the models don't scale up; they imply the need for too many neurons and synapses.

Stewart and Eliasmith make this argument in detail [7], using as examples Hummel and Holyoak's LISA model [4] and the Neural Blackboard Architecture (NBA) of van der Velde and de Kamps [8]. Both of these models dedicate specific neurons to represent specific entities such as symbols, propositions, and potential relations between them. Any entities that may potentially be bound have (possibly indirect) synaptic connections between them. The models differ in their binding mechanisms. LISA relies on temporal synchrony (bound entities are simultaneously active), while NBA relies on gating assemblies (analogous to telephone exchanges in allowing dynamic selection of connection paths).

If LISA held a human-sized vocabulary of concepts and relations, Stewart and Eliasmith estimate that it would need roughly as many neurons as the entire human brain in order to represent simple propositions like chase(dog,cat) and many more to represent higher order propositions like know(boy,chase(dog,cat)). NBA has greater re-use of physical resources than LISA, so Stewart and Eliasmith estimate a much lower number of neurons (approximately equivalent to all the language areas of the brain) to implement a similar scale-up. However, this only covers representation of linguistic structures without allowing any mechanisms to manipulate or use them.

The connectivity requirements are even more concerning than the number of neurons. LISA and NBA require that connection paths exist between the physical resources representing entities that may possibly be bound. Higher-order relations make the situation worse because representations of composite structures must be connected to representations of other composite structures. Analogy generates the worst scenario because novel correspondences are found between previously unrelated (possibly novel) composite structures. This implies the need for physical connections between everything that might possibly be represented.

1.3. A Cause for the Problems - Lack of Suitable Virtual Machines

We believe that the lack of functionality and lack of scaling both flow from the same source: the dedication of specific physical resources to represent specific entities. This issue can be addressed by the introduction of virtual machines (machines in which the causally effective components are not isomorphic to the physically implementing components). In a complex system one virtual machine may be implemented in terms of other virtual machines. This allows multiple levels of functional organisation such that higher levels can be implemented in multiple different ways and be isolated from the implementation details of lower levels. This separation of concerns is very helpful in a designed system because it allows cross-level interactions to be ignored.

Brains are not designed artefacts, but Sloman argues that evolution may have discovered the benefits of virtual machines [9]. However, BICA systems are designed artefacts, so the BI facet has to acknowledge the evolutionary origin of the inspiration while the CA facet respects the constraints of design. Sloman's view of biological virtual machines touches on Marr's analytical levels and the general concept of levels of abstraction but there is not the space to pursue these relations. Rather, we focus on the impact of virtual machines on scalability and ease of design.

Neural models are most naturally construed in terms of neurons, activations, and synaptic weights rather than symbols, rules and variables. This suggests a computational structure like an analogue computer. The output of each neuron represents some specific property of interest and may be fed to other neurons as input to further calculations. The label identifying what each neuron refers to is managed externally to the system (e.g. by the designer), is not available to the system, and plays no part in the calculation. Analogue computers are very ill suited to direct implementation of classical symbolic computation because the set of entities to be represented is dynamic, implying dynamic rewiring of the computer.

Given that we must implement on a neurally inspired substrate, what is needed is a suitable virtual machine to interpose between the neural hardware and the symbolic computation. It needs to provide functionality that is congenial to symbolic computation, and have a non-classical implementation on the neural substrate so that it avoids the scaling problems. Entities at the symbolic level must be freed from dedicated resources at the neural level. This will also simplify design by freeing the designer from having to micromanage the neural hardware to achieve symbolic ends.

2. On Designing a Cognitive Architecture

BICA is about "creating a real-life computational equivalent of the human mind" [10]. While cognitive neuroscience seeks only to explain cognition, BICA must also support the creation of cognitive systems. This design exercise is a human activity with human limitations that must be accommodated.

2.1. What I do not understand I cannot create[1]

In designing an artefact it is important that the number of interactions between the causal components be kept to a level that can be managed by the designer. If you can't understand the consequences of the interactions you can't predict the consequences of your designs and therefore you can't design. Sloman cites the benefits of virtual machines providing "vertical modularity" for both engineered and evolved systems [9]. Consequently, it is important that the virtual machines introduced in order to achieve the goals of functionality and scalability also simplify the conceptualisation of the BICA system so that it can be designed.

2.2. A Potential Tool Stack

In software engineering, a tool stack is a set of mutually compatible tools that allows the programmer to design and build a class of software systems. We are proposing a tool stack for building BICA systems. The aim is to have a set of mutually compatible tools that support the design and implementation of cognitive functionality on a neurally inspired substrate.

The tool stack consists of the Neural Engineering Framework (NEF), Vector Symbolic Architecture (VSA), and Data-Oriented Parsing (DOP). VSA provides the virtual machine that implements symbolic functionality without isomorphically mapping computational entities to physical resources. NEF provides a process for mapping the VSA abstract connectionist model, to a realistic neural implementation. DOP provides a symbolic/statistical mechanism for recognising and generating

[1] With apologies to Richard Feynman.

composite structures based on fragments of previously encountered composite structures. DOP can be viewed as a virtual machine that implements a compositional memory system on the substrate of symbolic computation provided by VSA. The tools may be ordered from NEF (most neural), through VSA, to DOP (most cognitive).

We describe our proposal as a potential tool stack because it has not yet been applied to a complete neurons-to-cognition design task. Each of the three tools has been extensively used in isolation. NEF and VSA have been used together extensively to implement cognitive functionality. However, the DOP/VSA interface does not exist and its development will require a serious research effort. Despite this incompleteness we believe it is valuable to describe this potential tool stack because even though it is not necessarily achievable or correct it appears to have the right characteristics and may inspire researchers to think of alternatives to the current flawed approaches.

The rest of this paper is devoted to description of the tool stack, starting with VSA.

2.2.1. Vector Symbolic Architecture

Vector Symbolic Architectures are a family of abstract connectionist models of symbolic computation. They are related to Smolensky's tensor product approach [11] but avoid its exponential resource scaling problem. VSAs are connectionist in that they are based on simple mathematical functions of vectors of scalar values (interpreted as the outputs of a set of neurons or as the weights of a set of synapses). They are abstract models in that they may contain neurally unrealistic features, such as bipolar outputs and multiplicative interactions at synapses.

VSA has peculiarities relative to typical connectionist approaches. It relies on the unintuitive mathematical properties of high-dimensional vector spaces (typically, thousands to tens of thousands of dimensions). The representations (patterns of activation or connection weights across the population of neurons or synapses) are generally distributed (i.e. there is generally no specific meaning attached to any neuron or synapse). There is a small set of fixed mathematical operators used to manipulate the vectors. VSA circuits consist of compositions of these operators. VSA circuits must be designed to implement the desired computation (unlike typical connectionist models where and the system learns the desired global function by adapting its synaptic weights). These peculiarities mean that the entire philosophy of the design and use of VSAs is quite different from typical connectionist models and familiarity with typical connectionist models may be more misleading than helpful. For a tutorial introduction to VSAs we recommend the extensive treatments of Kanerva [12] and Plate [13].

Some VSA properties relevant to the focus of this paper are: Representations generally behave like symbols (two vectors chosen at random are almost always nearly orthogonal) but graded representations can easily be constructed; Representations only require low resolution values of the underlying neurons and synapses – single bit precision is sufficient; Representations are very robust to noise and damage in the underlying neurons and synapses; The binding of two entities is represented by a vector of the same dimensionality as the vectors to be bound, so arbitrarily complex composite structures can be constructed in a fixed dimension vector space (to a limit imposed by signal to noise ratio considerations); Representations of discrete data structures such as trees and graphs can easily be constructed; Binding and unbinding are primitive operations requiring only a single pass through the relevant operator; The binding function is blind to the content of the values being bound, so composite structures can be used as values (or variables); Variables and values are both represented as values in the vector space, so both may be manipulated; The binding function is blind to the content of the values being bound so processing time is independent of the complexity of the structures being operated on; The underlying neurons do not establish any representationally privileged directions in the vector space, instead these are established by the contents of a cleanup memory, therefore the dimensions of the representation space can be dynamically changed by adding items to the clean up memory (like learning new concepts); Algebraic parallelism provides the computational effect of parallel processing by virtue of the underlying mathematics at no computational cost (in exactly the same way that 2*(3 + 4) = 2*3 + 2*4 = 2*7 provides two multiplications for the cost of one); Algebraic parallelism allows holistic processing, where composite structures may be manipulated without first decomposing them; The representation of a binding can be applied to other representations as a substitution operator; Substitution makes every value potentially a variable because other values can be substituted for it.

As a consequence of these and other properties, VSAs meet Jackendoff's challenges [14]. They also support virtualisation of the processing mechanisms. For example, a binding of two representations may be interpreted and used as a feature detector [15]. If the two entities bound represent the perceptual features of your grandmother and the label "*grandmother*" then querying the binding with the representation of the perceptual features of your grandmother will yield the label "*grandmother*". This is a virtual implementation of the infamous grandmother detector cell. It provides the functionality of the grandmother detector but doesn't require a dedicated physical resource, so it becomes much more plausible. Such detectors can be rapidly created with one-shot learning. The requirement to allow for anything to connect to anything is far less onerous when the connections are virtualised this way.

Gayler and Levy provide another example of the benefits of virtualisation and algebraic parallelism [16]. This system is a recurrent VSA circuit for calculating the isomorphism (i.e. structural match) between two graphs. Isomorphisms are usually found by discrete search techniques. However, it is possible to embed the problem in a continuous space and Pelillo has recast it as a problem of finding the maximum of a continuous function [17]. His method can easily be translated to a neural implementation where specific neurons correspond to specific entities, such as possible mappings between vertices of the graphs. The shortcoming of this approach is that the structure of the neural network is specific to the pair of input graphs. Consequently, every isomorphism problem requires the construction of a unique neural network. This would not be feasible if it were a component of a cognitive system.

The VSA version of this algorithm embeds the continuous optimisation problem in a much higher dimensional space where all aspects of the problem to be solved are represented as patterns of activation. Only one physical neural network is required to handle all possible input graphs. The VSA representations of specific graphs are virtual implementations of the neural network required to find the isomorphism.

Algebraic parallelism provides an analogous advantage for processing. The graph isomorphism process requires complex data structures such as a representation of all the possible mappings between the vertices of the two graphs. In a classical symbolic implementation this would require a process to iterate over the representations of the two graphs, with a computational cost proportional to the total number of possible vertex mappings. In the VSA version, binding the representation of the set of vertices of the first graph to the representation of the set of vertices of the second graph generates a representation of the set of all possible vertex mappings. This operation requires only constant time and constant cost, independent of the choice of graphs.

2.2.2. Neural Engineering Framework

The Neural Engineering Framework of Eliasmith & Anderson is "a synthesis of … ideas in computational … neuroscience, [combining elements of] neural coding, neural computation, physiology, communications theory, control theory, representation, dynamics, and probability theory" [18]. It is a method for taking a mathematical description of desired system behaviour and translating it to a network of realistic neural components that implements the behaviour.

The construction of a network from the specification of its behaviour gives NEF models a much more designed flavour than typical connectionist models, which rely on the optimisation of connection weights in order to achieve the desired functionality. Learning can be implemented in NEF as part of the desired functionality but it is not necessary for the construction of the network, whereas in a typical connectionist system learning is essential to the construction of the network. As a component of a cognitive neuroscience theory NEF requires a network construction mechanism such as ontogeny. However, as a design component in BICA no such mechanism is required and the comprehensible link from function to form is an advantage compared to reliance on weight learning which may or may not deliver the desired functionality.

For our purposes, the challenge of NEF lies in being able to write a mathematical description of the desired behaviour. It is not obvious how one could write a description of cognitive behaviours adequate to meet Jackendoff's challenges. However, the primitive operations of VSA can be easily specified in a mathematical form acceptable to NEF. This has been done by Eliasmith, for example, to implement a realistic neural model of the Wason task [19]. VSA provides suitable symbolic primitives for a simple implementation of the cognitive task and NEF derives an implementation of the VSA circuit onto biologically realistic spiking neurons. The fixed architectures designed by NEF to implement VSA systems are also capable of learning (e.g. rule induction from Raven's Progressive Matrices [20]).

2.2.3. Data-Oriented Parsing

VSA provides symbolic functional primitives that need to be composed into a fixed circuit to implement a specific symbolic computation (e.g. the Wason task or graph isomorphism). It is possible that cognition requires a large number of different VSA circuits to implement many specialised symbolic computations. However, if there does exist some generic symbolic computation that is the basis for a range of cognitive tasks it would be very advantageous to develop a VSA implementation. We believe that Data-Oriented Parsing may be such a generic computation.

DOP is a mechanism for recognising and generating compositional structure. It was developed in computational linguistics as a model of language perception and production, driven by representations of concrete language experiences rather than abstract grammatical rules [21]. In DOP, grammar is an emergent functional property of the statistical structural constraints implicit in a corpus of concrete experiences.

Assume a corpus of parsed utterances (i.e. utterances annotated with compositional structures relating to their interpretation). This is then decomposed into structural fragments that may be recomposed later to yield parses of novel utterances (e.g. the components of "*John loves Mary*" may be recomposed to yield "*Mary loves John*").

DOP considers all recompositions of the stored fragments consistent with the utterance, calculates the probability of each recomposition, sums over all derivations that yield the same parse, and chooses the most probable parse.

Although DOP is usually described in terms of linguistic representations it is actually applicable to any compositional representations that can be decomposed into fragments and recomposed. It has been applied to multiple linguistic syntactic representations, semantic representations, music, and physics problem solving [22]. It has also been used unsupervised to learn, how to structure its input [23]. If a structurally generated utterance is parsed randomly, the statistical relations between the fragments of the random parse capture the regularities of the utterances.

DOP appears to be a good candidate for implementation of a compositional memory for recognising novel stimuli as novel compositions of familiar components. There is not yet a VSA implementation of DOP, but good reason to hope it is possible. The DOP algorithm is very simple and uniform. The reliance on concrete fragments makes it look very like a memory system. The recomposition process can be construed as composition by substitution, which is a primitive operator in VSA. The probabilistic calculation to select the best parse is similar in spirit to the optimisation process used by the VSA graph isomorphism circuit. The computationally expensive probability calculation looks like it would be amenable to the algebraic parallelism of VSA.

3. Conclusion

We have argued that BICA must simultaneously be compatible with the explanation of human cognition and support the human design of artificial cognitive systems. Most cognitive neuroscience models fail to provide a basis for implementation because they neglect necessary levels of functional organisation in jumping directly from physical phenomena to cognitive behaviour. Of those models that do attempt to account for the intervening levels, most either fail to implement the required cognitive functionality or fail to scale adequately.

We argue that the problems in the credible neural models arise because of identifying computational entities with physical resources such as neurons and synapses. This issue can be avoided by introducing suitable virtual machines to break the direct correspondence between computational and physical entities.

We have proposed a tool stack that supports the required functionality and scaling by providing suitable virtual machines and simultaneously simplifies the human design of cognitive architectures by providing vertical modularity. This tool stack leaves many important research questions untouched. It is neutral with respect to issues such as: functional specialisation of computations and their distribution across neuroanatomy; the broad structure of the computational architecture; the details of what is to be represented and how. However, the simplification of design should make it easier to investigate these questions. This specific tool stack may not be the best possible but it provides an example of what is possible and a foundation for further development.

Acknowledgments

We thank Chris Eliasmith and the anonymous reviewers for their helpful comments.

References

[1] Richard Feynman (n.d.). In Wikiquote. Retrieved April 17, 2010, from http://en.wikiquote.org/wiki/Richard_Feynman

[2] Lin, L., Osan, R., & Tsien, J. Z. (2006). Organizing principles of real-time memory encoding: neural clique assemblies and universal neural codes. *Trends in Neurosciences*, *29*, 48-57.

[3] Ananthanarayanan, R., Esser, S. K., Simon, H. D., & Modha, D. S. (2009). The cat is out of the bag: Cortical simulations with 10^9 neurons and 10^{13} synapses. *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*. Portland, OR: ACM Press.

[4] Hummel, J. E., & Holyoak, K. J. (2003). A symbolic-connectionist theory of relational inference and generalization. *Psychological Review*, *110*, 220-264.

[5] Jackendoff, R. (2002). *Foundations of language: Brain, meaning, grammar, evolution.* Oxford, UK: Oxford University Press.

[6] Barsalou, L. W. (1999). Perceptual symbol systems. *Behavioral and Brain Sciences, 22*, 577-660.

[7] Stewart, T., & Eliasmith, C. (in press). Compositionality and biologically plausible models. In M. Werning, W. Hinzen, & E. Machery (Eds.), *Oxford Handbook of Compositionality*. Oxford, UK: Oxford University Press.

[8] van der Velde, F., & de Kamps, M. (2006). Neural blackboard architectures of combinatorial structures in cognition. *Behavioral and Brain Sciences*, *29*, 37-70.

[9] Sloman, A. (2009). What cognitive scientists need to know about virtual machines. *Proceedings of the 31st Annual Conference of the Cognitive Science Society* (pp. 1210-1215). Austin, TX: Cognitive Science Society.

[10] BICA-2010 Call for Papers (2010). Retrieved August 31, 2010, from http://bicasymposium.com/bica2010_cfp.pdf

[11] Smolensky, P. (1990). Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence, 46*, 159-217.

[12] Kanerva, P. (2009). Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors. *Cognitive Computation*, *1*, 139-159.

[13] Plate, T. A. (2003). *Holographic reduced representation: Distributed representation for cognitive science*. Stanford, CA: CSLI Publications.

[14] Gayler, R. W. (2003). Vector Symbolic Architectures answer Jackendoff's challenges for cognitive neuroscience. *Proceedings of the ICCS/ASCS International Conference on Cognitive Science* (pp. 133-138). Sydney, Australia: University of New South Wales.

[15] Gayler, R. W., & Wales, R. (1998). Connections, binding, unification, and analogical promiscuity. *Advances in analogy research: Integration of theory and data from the cognitive, computational, and neural sciences* (pp. 181-190). Sofia, Bulgaria: New Bulgarian University.

[16] Gayler, R. W., & Levy, S. D. (2009). A distributed basis for analogical mapping. *New frontiers in analogy research; Proceedings of the Second International Analogy Conference - Analogy 09* (pp. 165-174). Sofia, Bulgaria: New Bulgarian University.

[17] Pelillo, M. (1999). Replicator equations, maximal cliques, and graph isomorphism. *Neural Computation*, *11*, 1933-1955.

[18] Eliasmith, C., & Anderson, C. H. (2003). *Neural engineering: Computation, representation and dynamics in neurobiological systems*. Cambridge, MA: MIT Press.

[19] Eliasmith, C. (2005). Cognition with neurons: A large-scale, biologically realistic model of the Wason task. *Proceedings of the 27 th Annual Meeting of the Cognitive Science Society*. Stresa , Italy: Lawrence Erlbaum Associates.

[20] Rasmussen, D., & Eliasmith, C. (2010). A neural model of rule generation in inductive reasoning. *Proceedings of the 32nd Annual Conference of the Cognitive Science Society* (pp. 61-66). Portland, OR: Cognitive Science Society.

[21] Bod, R. (1998). *Beyond grammar: An experience-based theory of language*. Stanford, CA, CSLI Publications.

[22] Bod, R. (2008). The Data-Oriented Parsing approach: Theory and application. In J. Fulcher & L. Jain (Eds.), *Computational intelligence: A compendium*. Berlin: Springer.

[23] Bod, R. (2009). From exemplar to grammar: A probabilistic analogy-based model of language learning. *Cognitive Science*, *33*, 752-793.