



Universiteit  
Leiden  
The Netherlands

## **Towards a constructivist methodology: Learning constructions by integrating in situ representations and productivity**

Van, der Velde F.

### **Citation**

Van, D. V. F. (2013). Towards a constructivist methodology: Learning constructions by integrating in situ representations and productivity. *Journal Of Artificial General Intelligence*, 3(3), 101-121. Retrieved from <https://hdl.handle.net/1887/78125>

Version: Not Applicable (or Unknown)

License:

Downloaded from: <https://hdl.handle.net/1887/78125>

**Note:** To cite this publication please use the final published version (if applicable).

# Towards a constructivist methodology: learning constructions by integrating in situ representations and productivity

**Frank van der Velde**

F.VANDERVELDE@UTWENTE.NL

*Technical Cognition, CTIT*

*University of Twente*

*P.O. Box 217*

*Enschede, 7500 AE, The Netherlands*

*IOP, Leiden University*

*Leiden, 2333 AH, The Netherlands*

**Editors:** Kristinn R. Thórisson, Eric Nivel, Ricardo Sanz

## Abstract

The ability to learn constructions may be important for the development of a self-organizing architecture for artificial general intelligence. Constructions are structural relations between more specific or more abstract conceptual representations. They can be derived from the processes of alignment, collocations and distributed equivalences. An architecture that integrates in situ grounded representations with cognitive productivity is ideally suited to learn constructions. This paper described such an architecture, based on neuronal assembly structures and neuronal 'blackboards' for grounded compositional representations. The paper outlines how constructions could be learned in such an architecture and how the architecture could eventually develop into an autonomous self-organizing architecture for artificial general intelligence.

**Keywords:** Constructions, in situ representations, cognitive productivity, compositional structures, neuronal assemblies, neural blackboards

## 1. Introduction

The aim of this paper is to argue that the ability to learn constructions (e.g., as based on construction grammar) may be an important characteristic of a self-organizing architecture for artificial general intelligence. The term "constructions" should be distinguished from the dichotomy between constructionist versus constructivist approaches to artificial general intelligence (Thórisson (2009)). The term as I use it here derives from a theory of the nature of human language (e.g., Goldberg (1995)) and how language is acquired (e.g., Goldberg (2006)). Linguistic constructions are in fact on the constructivist side of the constructionist versus constructivist dichotomy.

Why would learning linguistic constructions be important for a self-organizing architecture for artificial general intelligence? First, because the ability to acquire language will be important for such an architecture. It is difficult to see how an autonomous intelligent system could exist without language. Also, constructions offer a means to learn about relations between events, which might also be used by an autonomous system to reason about its own abilities.

The theory of linguistic constructions, or construction grammar, arose (in part) as a reaction to the theory of universal grammar developed by Chomsky. According to universal grammar, humans are born with a set of universal rules for language (or with the ability to learn precisely these rules).

The inborn nature of the rules is reflected by their universal nature. All humans possess the same set of rules, which they use for their own language. The differences between natural languages are superficial. Universal grammar accounts for these differences by positing transformation rules. The underlying form of a sentence, which is the same for all languages, is transformed by the transformation rules in the different spoken forms as found in different languages. Conversely, understanding a sentence derives from retrieving the underlying sentence form from the different spoken forms offered by different languages. Universal grammar is on the constructionist side of the constructionist versus constructivist dichotomy. If we want a generally intelligent system to have natural language, we need to give it the rules for the universal grammar, or the program that would allow the system to learn these rules and no other.

Construction grammar does not see the differences between languages as superficial. Instead, they reflect the different learning environments that users of natural language have experienced. This also accounts for the differences between dialects of the same language. When a speaker is confronted with a particular language in early life, he or she will learn the structure of that language. As a result, languages can have different structures as they develop in relatively isolated communities. It is important to note that the experience of a language learner not just consists of the language he or she hears, but also of the events that occur in the environments. The relations between these events, such as the relation between an agent and an action, are then expressed in linguistic form depending on the given language at hand. For example, the agent could be expressed as a subject and the action as a verb, or the action could be expressed as a verb and the subject as a modification of the verb. In this way, it is possible to translate one language into another (although subtleties may be lost in translation).

To illustrate the difference between the universal grammar (or similar constructionist approaches to language) and construction grammar, consider the verb (predicate) *give*. In universal grammar a verb like *give* obeys a set of syntactic rules. In particular, *give* has three arguments that have to be expressed in a sentence with *give*. So, we could say *John gives Mary a book* or *John gives a book to Mary*. These two sentences are superficially different, but they express the same predicate relation in which an object is willingly handed over by one person to another. The transformation rules of English transform each of the two different superficial forms into this underlying predicate relation.

In the view of construction grammar the sentences *John gives Mary a book* or *John gives a book to Mary* are different, because they express two different constructions. The first sentence is a di-transitive construction in which the verb *give* has two objects. In the second sentence, the verb *give* has one object and a complement introduced by *to*. Because both sentences are based on different constructions, there is no need for transformation rules. Construction grammar does not assume that a sentence has a surface structure and a more fundamental structure underneath. Instead, it takes sentences at their face value, and it assumes that we learn a particular construction because it reflects a particular way we communicate a message. In the case of the verb *give*, we have two constructions because they reflect different forms of meaning or use. For example, there is focus on the given object in *John gives Mary a book*, whereas the focus is on the receiver in *John gives a book to Mary*. Also, in the di-transitive construction the receiver has to be animate, as in *John brought Mary a drink* (Goldberg (1995)). This is not the case in the *to* construction, as illustrated by the comparison between *John brought a drink to the table* and the infelicitous sentence *John brought the table a drink*.

Constructions thus include all aspects of a relation between objects or events that are somehow meaningful and which influence either the syntax of a sentence, its meaning (semantics) or its

pragmatics. We learn these constructions because over time we notice similarities in the relations between objects or events (as the similarity between verbs like *give* and *bring* in the di-transitive or *to* constructions). Because language and communication are needed for general intelligence, perhaps also for self-reflection (which also seems to be some form of communication), the ability to learn constructions could be important for artificial general intelligence.

Construction grammar not only has roots in language but also in cognitive science and psychology. In the next section I will briefly discuss these roots. Then, I will discuss the learning process that could underlie learning constructions.

## 2. Constructions and constructivism

Construction grammar has a long history in linguistics as an alternative to the mainstream view of generative grammar, dating back (at least) to Fillmore (1965). But it is also related to constructivist views in cognitive science (e.g., Lakoff (1987), developmental psychology (e.g., Piaget (1950)), and learning theory and cybernetics (e.g., Thompson (1995)). In this view, interacting with the environment we develop constructs that provide meaning about our environment. In particular, a perceived discrepancy between the world view of the learner and the world view of others or (new) experiences can trigger the development of new constructs. In human development, this process will also depend on the child's stage of development (Piaget (1950)).

The notion that our concepts are developed in interaction with the world is clearly related to the idea of grounding (e.g., Harnad (1991)), and the development of constructions, also under the influence of perceived discrepancies, is related to notions of development in the construction grammar approach (Goldberg (2006)). It is beyond the scope of this article to discuss these relations in detail, but one question deserves attention. Why focus on construction grammar instead of the more general views of constructionism?

Firstly, because construction grammar by now provides a descriptive theory of linguistic phenomena that matches that of other (full blown) linguistic theories (e.g. Culicover and Jackendoff (2005)). This provides us with a rich and detailed (and empirically validated) set of (linguistic) phenomena that can serve as targets for a self-organizing architecture of intelligence. Moreover the phenomena can be distinguished in their complexity and (hierarchical) structure, which offers the potential for a step by step development of such architectures. I am not aware of such a rich and detailed set of examples in more general constructivist theories.

Secondly, language will be an important feature of any high-level self-organizing architecture of intelligence. Of course, numerous animals operate successfully in their environments without using (complex forms of) language. But for any high-level form of self-organizing architecture of general intelligence, language is indispensable. It is the best way for such a system to process and generate information beyond the here and now. Indeed, due to language we can share in the knowledge domains created over centuries. It is hard to see how any self-organizing architecture of intelligence could do all of that by its own. In fact, I am always a bit surprised to hear (!) or read (!) opinions that language is not crucial for high level intelligence. How could you ever express or even develop such an opinion without using language?

Thirdly, although language may be more limited than cognition in general, they are closely related. It is through language that we express and (partly) shape our ideas and opinions. This suggests that language is somehow rooted in our more general cognitive abilities. Indeed, the notion of in situ grounded representations is based on this notion (see below). So, by learning

to develop linguistic constructions, we might come to understand better how more general cognitive constructions could develop as well.

### 3. Learning constructions

To see how an artificial general intelligent system could learn constructions, consider a robot that has to operate in an unfamiliar environment. The robot could begin to learn structural relations between the objects in its environment by navigating the environment and using perception and attention to select objects. Initially, it will develop sequences of representations of the objects it has encountered, without yet any representation of relations between these objects or its own actions. Using these sequences it can then begin to learn structural relations based on the processes of alignment and comparison (Edelman (2008)).

In linguistics, alignment and comparison have been proposed as mechanisms to learn basic syntactical structures from a corpus of sentences. Initially, these sentences are just sequences of words, but with the repeated process of alignment and comparison these sequences can be transformed into structural representations that reflect the syntactic relations in the sentences (Harris (1968)). The means to do this are given by two complementary learning mechanisms based on alignment and comparison: collocations and distributed equivalences. The collocations (or syntagmatic regularities) are (partial) structures or words that a set of sentences have in common, such as the verb *give* in all sentences of the form *X gives Y Z* or sentences of the form *X gives Z to Y*. So, collocations determine the basic structure of a construction.

Distributional equivalences (or paradigmatic regularities) are in a way the opposite from collocations. Distributed equivalences are the structural similarities between the words that can fill (qualified) variables. Distributed equivalences could be very abstract, such as the notion that in a sentence like *John sees X* the *X* slot could be filled with any word referring to an object. But distributed equivalences can also be more restricted. For example, as noted earlier, the receiver has to be animate in the di-transitive construction *X gives Y Z*, but it can be inanimate in the to construction *X gives Z to Y*. We learn this difference because we encounter the di-transitive construction only in events where an animate agent interacts with an animate receiver. In the di-transitive construction itself, the verb *give* could form a distributed equivalence with other verbs that form di-transitive relations, such as the verb *bring*.

The constructions we learn with collocations and distributed equivalences can thus consist of relations between variables, indicated by abstract concepts (e.g., nouns, verbs, animate objects). But they can also consist of relations between variables and specific words (e.g., *send X to the cleaners*), or relations between specific words as found in idioms (e.g., Goldberg, 2006). This also marks a difference between universal grammar and construction grammar. Universal grammar posits a distinction between rules (syntax) and words (lexicon). The ‘real program’ for language is found in the syntax, the basis of which is inborn. The lexicon is just a collection of words learned in life. Construction grammar does not make this distinction between program and content. We learn words and relations in the same way and store them in a similar manner in memory.

Given the similarities between linguistic structures and structures in the visual world (e.g., Jackendoff and Pinker (2005)), collocations and distributed equivalences could also be used for detecting constructions in an environment. When a robot navigates its environment it could align two (or more) object or event sequences on the basis of the similarity of the objects or events they contain. For example, if one navigation sequence consists of the objects A G D E and another of the

objects G D F H, then an alignment between these two sequences would align the second and third object of the first sequence (G D) with the first and second object of the second sequence.

The aligned sequences then form the basis for detecting collocations and distributional equivalences. Collocations consist of sequence parts that are similar in a set of sequences. The sequence part G D in the sequences above is an example. Or, assume that a robotic system frequently encounters a set of sequences like G D ? H, in which the objects G, D and H are (frequently) present at the first, second and fourth position, and the third position is filled by a variable object. Then G D - H form a collocation.

The (gradual) emergence of a collocation suggests the existence of a structural entity in the environment (Barlow (1990)), that is, a construction. A new representation can then be formed that tags this collocation in the structural representation of the environment. For example, in a house or office environment, the legs below a tabletop form a collocation, so does combination of legs and tabletop. These collocations can form the basis of a new representation that can be used to embed the object (table) in a representation that describes the relations between the table and other objects in the room.

As noted, distributional equivalences are in a way the opposite from collocations. For example, in the set of sequences G D A H, G D B H and G D C H, the objects A, B and C form a set of different objects that frequently occur in the same context (i.e., the collocation G D - H). Apparently, the common context provided by the collocation reveals some of the characteristics that these objects have in common, such as the way they can be used in a given environment. Examples in a house or office environment could be the objects that one can find on tables or desks. Typically, these objects are not found on chairs or on the floor. Such distributional equivalences form (more abstract) categories of objects, which are useful for understanding functional relations between objects.

Thus, collocations and distributed equivalences would seem to suffice for the development of constructions because they can identify the relations that form constructions and the (kind of) words or concepts that could fill the variable slots (if any) that occur in these constructions. In contrast with program based theories of language (or even cognition in general), such as universal grammar, there is no real distinction between constructions at an abstract level and constructions at an idiomatic level, and the variants in between. Each of these would arise by the combined processes of collocations and distributed equivalences. The abstraction level of a construction is determined by the nature of the distributed equivalences it contains, which in turn is determined by the environment (collocation) in which the concepts that form a distributed equivalence occur.

The loss of the distinction between a program and the memorized items on which the program operates seems to be a promising basis for the development of a self-organizing architecture for an autonomous general intelligent system. There is no need to write a program that covers all relevant aspects of a cognitive domain, such as universal grammar in the case of language. Instead, the autonomy of the system will increase when it is capable of forming more constructions. The construction based nature of human language supports this view. The human child learns natural language because it can form constructions based on its experience. These constructions start at a basic level and later on develop into hierarchical layers of organization (e.g., Goldberg (2006)), which supports the bootstrapping nature of self-organization.

But we need an architectural basis to begin with. Given the importance of constructions for human language (and cognition in general) this architectural basis would have certain features in common with the human mind/brain. In the next section I will propose such an architecture.

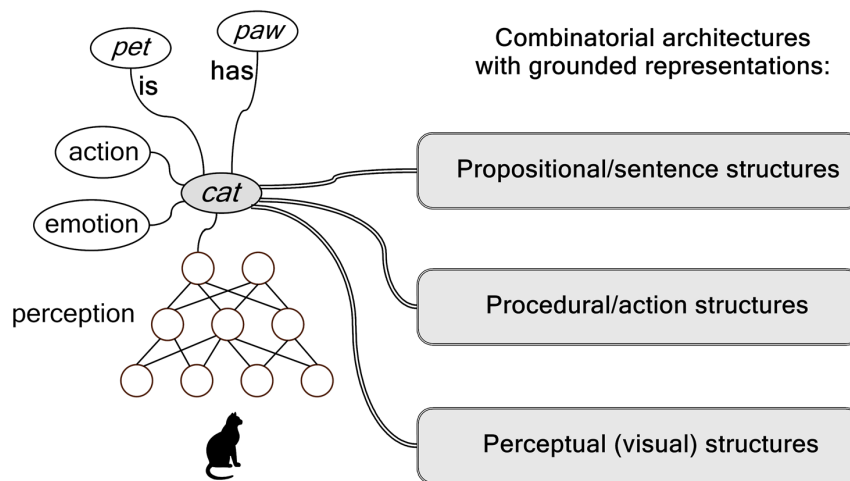


Figure 1: Grounded *in situ* representations are embedded in several specialized architectures, needed for specific cognitive processing based on specific combinatorial structures.

#### 4. Architecture based on in situ representations

The architecture aims to integrate two features of the human mind/brain: grounding of representations and cognitive productivity as found in the ability to learn, create and process compositional structures. Examples of these structures are sentences composed of words, sequences of actions or visual scenes composed of visual features (e.g., shape, color, location) and their relations.

The integration of grounding and productivity imposes constraints on the conceptual representations that can occur in compositional structures. Grounding entails that the cognitive representations are derived from the processes (e.g., perception, action, emotion) that give rise to the representations, thereby giving meaning to these representations. The representations then remain ‘in situ’ when they are part of more complex compositional representations such as structures. That is, they cannot be copied and transported like (symbolic) representations in digital computing systems. Instead, in situ representations share characteristics with the neuronal representations as proposed by Hebb (1949). In Hebb’s view, cognitive representations in the human brain consist of network structures, or ‘neuronal assemblies’, that are distributed over the brain (potentially over wide areas). They develop over time and are continuously modified by experience.

Figure 1 illustrates a neuronal assembly that instantiates the grounded representation of the concept *cat*. It grounds the concept *cat* by interconnecting all aspects related to cats. For example, it includes all perceptual information about cats, as, e.g., given by the neural networks in the visual cortex that classify an object as *cat*, or the networks that identify a spoken word as *cat*. It also includes action processes related to cats (e.g., the embodied experience of stroking a cat, or the ability to pronounce the word *cat*), and information (e.g., emotions) associated with cats. But in contrast with Hebbian assemblies, the assemblies in Figure 1 are not only associative. They can also include relational, semantic or syntactic information, as illustrated with the relations *cat is pet*

and *cat has paw*. Relational, semantic or syntactic forms of information, such as *cat is pet*, do not derive from associations between assemblies (e.g., between *cat* and *pet*).

The neuronal assembly in Figure 1 forms an in situ representation of the concept *cat*, because it is not (and cannot) be copied or transported to form compositional structures such as sentences or relations. To achieve compositional forms of processing with in situ representations they have to be embedded in specific (modular) neuronal architectures to form and process compositional structures. For example, the activation of the relations illustrated in Figure 1 depends on the 'conditional connections' between the assemblies involved (van der Velde and de Kamps (2006)). Conditional connections can be activated when certain conditions are met, as indicated by the labels *is* and *has* in Figure 1. These conditions can result from contextual information or from specific information, e.g., as given by questions. For example, the query *cat is?* activates the condition *is in* in Figure 1.

Conditional connections can be instantiated with specific neuronal circuits (van der Velde and de Kamps (2006)). These circuits belong to modular neuronal architectures that can create and process specific forms of compositional structures (e.g., sequential, semantic, syntactic, phonological, complex actions). The in situ representations as illustrated in Figure 1 are embedded in these architectures (depending on the nature of the concept involved). Interactions between the architectures proceed through the in situ representations (neuronal assemblies) they share.

#### 4.1 Grounding in compositional structures

To understand the constraints given by the integration of grounding and cognitive productivity (compositional structures) it is useful to compare architectures that integrate cognitive productivity with grounded representations with cognitive architectures based on symbolic computing (or symbol manipulation). Examples of the latter are the cognitive architectures described by Anderson (1983) and Newell (1990). Symbolic architectures achieve productivity by using symbol manipulation to process or create compositional structures. Symbol manipulation depends on the ability to make copies of symbols and to transport them to other locations. Newell (1990) described this process as follows: "When processing *The cat is on the mat ...* the local computation at some point encounters *cat*; it must go from *cat* to a body of (encoded) knowledge associated with *cat* and bring back something that represents that a cat is being referred to, that the word *cat* is a noun (and perhaps other possibilities), and so on" (p. 74, italics added). Thus, a symbol for the word *cat* is copied and pasted into a representation of *cat on the mat*. Because symbols can be copied and transported, they can be used to create complex compositional structures such as propositions, with symbols as constituents. The digital computer, with a CPU and symbolic representations is the computational architecture needed to implement these processes (e.g., Fodor and Pylyshyn (1988)).

The capacity of symbolic architectures to store (represent) and process complex compositional (e.g., sentence-like) information far exceeds that of humans. But interpreting information in a way that could produce meaningful answers or purposive actions is far more difficult with these architectures. In part, this is due to the ungrounded nature of symbols, which, in turn, is partly due to the fact that symbols are copied and transported in these cognitive systems.

The ungrounded nature of symbols in cognitive architectures has been criticized by cognitive scientists (e.g., Harnad (1991), Barsalou (1999)). To improve this situation, cognitive models have been developed that associate labels for words (e.g., nouns, adjectives, verbs) with perceptual and behavioural information (e.g., Roy (2005)). For example, colour words can be associated with

perceptual representations of colours (Mojsilovic (2005)), or verb semantics can be associated with action control sequences (Roy (2005)).

Models like these are a significant improvement over the use of arbitrary symbols to represent concepts. However, the conceptual representations and their associations with perceptual representations are stored in long-term memory in these models. To form compositional structures with these concepts (e.g., *cat on the mat*), the conceptual representations still have to be copied from long-term memory and pasted into the compositional structure at hand, as in Newell's cognitive architecture. Thus, the direct grounding of the conceptual representations is lost when they are used as constituents in compositional structures.

An active decision is then needed to retrieve the associated perceptual information, stored somewhere else in the architecture. This raises the question of who (or what) in the architecture makes these decisions, and on the basis of what information. Furthermore, given that it takes time to search and retrieve information, there are limits on the amount of information that can be retrieved and the frequency with which information can be renewed (van der Velde (2010)). These difficulties seriously affect the ability of such cognitive architectures to adaptively deal with information in complex environments. It hampers, for example, fast processing of ambiguous information in a context-sensitive way (van der Velde and de Kamps (2011)).

The use of 'pointers,' as in certain programming languages (e.g., C++), does not eliminate these difficulties at all. A pointer does not carry any conceptual information, but only refers to an address at which (potentially) conceptual information is stored. Again, an active decision is needed to obtain that information, which raises the same issues as outlined above.

To avoid these problems, grounded representations have to remain grounded when they are parts of compositional structures. Cognitive representations as illustrated in Figure 1 satisfy this constraint because they are grounded and they are used directly in compositional structures, that is, they remain in situ in compositional structures. The in situ nature of these representations is in fact stronger than, and entails, grounding. As Figure 1 illustrates, grounding results from the association of a conceptual representation with, e.g., perceptual, action, and emotional representations that are activated in the process of learning the concept. This grounding is inherent in an in situ representation as given by a neuronal assembly. But, as noted above, when an initially grounded symbol is copied and stored elsewhere in a symbolic architecture, the immediate and direct grounding of a copied symbol is lost, because it does not constitute an in situ representation.

## 4.2 Constraints for in situ representations in compositional structures

The fact that grounded in situ representations cannot be copied and transported raises the question of how they can be used to represent and process compositional cognitive structures, which form the basis of the productivity of complex (human) cognitive processing. Given the fact that neuronal assemblies as illustrated in Figure 1 are in situ representations, this question is similar to the issue of how complex cognitive processes (or 'thinking') could be based on (in situ) neuronal (cell) assemblies.

Hebb (1949) addressed this issue. In his view, the sequential activation of cell-assemblies, or 'phase sequence', forms the basis of high-level cognition (p. xix): "A series of such events [i.e., activated cell assemblies] constitutes a 'phase sequence' - the thought process. Each assembly action may be aroused by a preceding assembly, by a sensory event, or-normally-by both. The central facilitation from one of these activities on the next is the prototype of 'attention'."

The in situ representations as illustrated in Figure 1 consist of neuronal assemblies. Hence, representation and processing compositional cognitive structures would consist of selectively activating (part of) the underlying neuronal assemblies. This selective activation is a form of attention. The activated assembly (or attended concept) dominates the neuronal dynamics (or cognitive process), until a new assembly is selectively activated, either by a new stimulus or by the preceding sequence of assemblies (or by both).

If the processing of compositional cognitive structures could result from selectively activating sequences of neuronal assemblies, each of the cognitive representations involved would be in situ (and thus grounded) throughout the cognitive process. This would eliminate the need for retrieving information about a concept stored elsewhere, e.g., by using a symbol as outlined by Newell (1990), because all conceptual information is available (i.e., can be activated) when the underlying neuronal assembly is active. The activation of additional information directly depends on the dynamics of the process at hand. In this way, all problems associated with retrieving additional information about a concept (e.g., who/what decides to retrieve new information; on the basis of what information; how often) are eliminated.

However, the neuronal instantiation of compositional cognitive structures has to satisfy a number of constraints. These constraints derive from the nature of compositional processing, as found, for example, in human linguistic processing. Jackendoff (2002), see also Marcus (2001) analysed the most important theoretical problems (challenges) that the compositional nature of language, and of cognition in general, presents to theories of neurocognition. These problems also pertain to architectures based on in situ representations. They are the massiveness of the binding problem, the problem of multiple instances (or the “problem of 2”) and the problem of variables.

The massiveness of the binding problem concerns the way in which the parts (or constituents) of a compositional structure are (temporarily) bound in a manner that preserves the relations between the constituents in the compositional structure. In vision, examples are found in the spatial composition of objects in a visual scene. In language, words are (temporarily) bound in a sentence structure. If words are represented in the brain by neuronal assemblies, as illustrated with *cat* in Figure 1, then these assemblies are bound when the words form a sentence. The binding has to preserve the relations between the words in the sentence. So, *cat* is bound as the subject to the sentence structure of *The cat is on the mat*, whereas *mat* is bound as the subject to the sentence structure of *The mat is on the cat*.

The binding problem is massive because of the productivity of human cognition. Humans can produce or understand a practically unlimited amount of sentences in natural language (Pinker (1998)). This precludes the possibility of representing each of these sentences by a dedicated neuronal circuit. Instead, most of these sentences will be represented by temporarily binding its words (i.e., neuronal ‘word’ assemblies) in accordance with the sentence structure.

The massiveness of the binding problem is enhanced because of the complex and hierarchical nature of many compositional structures. For example, sentences can have relative clauses as in *The cat on the mat is red*. Here, *cat* is bound as subject to the phrase *The cat on the mat*, and this whole phrase is bound as subject to *The cat on the mat is red*. These complex and hierarchical binding relations preclude a general solution in terms of activation labeling, such as synchrony of activation (van der Velde and de Kamps (2006)).

The problem of multiple instances (or the “problem of 2”) is directly related to the in situ nature of neuronal representation. For example, in the sentence *The little star is beside the big star* (Jackendoff (2002)), the word *star* occurs twice. If *star* is represented by a neuronal assembly

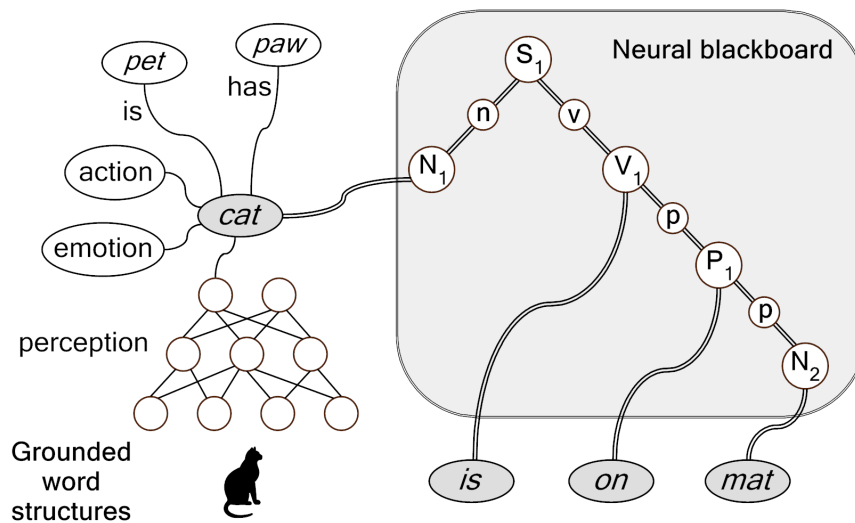


Figure 2: Illustration of the combinatorial structure *The cat is on the mat* (ignoring *the*), with grounded in situ representations for the words. The circles in the neural blackboard represent populations and circuits of neurons. The double line connections represent conditional connections. (N, n = noun; P, p = preposition; S = sentence; V, v = verb.)

(in situ representation) as in Figure 1, how are both copies of *star* represented, with the first copy bound to *little* (and bound as subject) and the second copy bound to *big* (and bound to the preposition phrase of the sentence)?

The problem of variables derives from the systematic nature of (high level) cognition. Specific relations can be instantiated on the basis of specialized neural circuits, such as the relation *cat has paw* or *cat is pet* in Figure 1. But systematic relations are represented with variables, such as the relation between *give(X,Y,Z)* and *own(Y,Z)*. Hence, the structure of the predicate *give* is represented in terms of variable slots, which can be filled with (bound by) arbitrary concepts (including sentences, as in *John gives Mary the cat on the mat*).

Cognition is systematic (Fodor and Pylyshyn (1988)), in the sense that one can learn from specific examples, e.g., *John gives Mary a cat*, thus *Mary owns a cat*, and apply that knowledge to all examples of the same kind, e.g., *give(X,Y,Z)* thus *own(Y,Z)*. Given the importance of systematicity for human cognition, this raises the question of how variable binding can be instantiated with in situ representations, such as neuronal assemblies in the brain.

### 4.3 Compositional structures based on situ representations

A solution of the problems discussed above is illustrated in Figure 2. This figure shows that grounded in situ representations of the words *cat*, *is*, *on*, and *mat* can be used to create a compositional structure of the sentence *The cat is on the mat* (ignoring *the*). The structure is created by forming temporal interconnections between the in situ representations (neuronal assemblies) of *cat*, *is*, *on*, and *mat* in a “neural blackboard architecture” for sentence structure (van der Velde and de Kamps (2006)).

The “neural blackboard” consists of neuronal ‘structure’ assemblies that (in the case of sentences) represent syntactical type information, such as sentence ( $S_1$ ), noun phrase ( $N_1$  and  $N_2$ ), verb phrase ( $V_1$ ) and prepositional phrase ( $P_1$ ). To create a sentence structure, assemblies representing specific syntactical type information (or syntax assemblies, for short) are temporarily connected (bound) to word assemblies of the same syntactical type. Binding is achieved with dedicated neural circuits. They constitute the ‘conditional’ connections and control networks, needed to implement relational structures (van der Velde and de Kamps (2006), van der Velde and de Kamps (2010)).

Thus, *cat* and *mat* are bound to the noun assemblies  $N_1$  and  $N_2$ , respectively. In turn, the syntax assemblies are temporarily bound to each other, in accordance with the sentence structure. So, *cat* is bound to  $N_1$ , which is bound to  $S_1$  as the subject of the sentence, and *is* is bound to  $V_1$ , which is bound to  $S_1$  as the main verb of the sentence. Furthermore, *on* is bound to  $P_1$ , which is bound to  $V_1$  and  $N_2$ , to represent the prepositional phrase *is on mat*.

Figure 3 illustrates how the neural blackboard architecture solves the problem of multiple instances (or “problem of 2”). Instead of making a copy, as in a symbolic representation of this sentence, the in situ representation of *cat* is connected twice to the sentence structure of *The cat is on the cat* in the neural blackboard for sentence structures. This can be achieved because the in situ grounded structure for *cat* is now bound to two noun assemblies,  $N_1$  and  $N_2$ . Because  $N_1$  is bound to  $S_1$ , *cat* is the subject of the sentence. But *cat* is also the noun in the prepositional phrase because  $N_2$  is bound to  $P_1$ . Both representations of *cat* in the sentence remain grounded and in situ in this way.

The neural blackboard architecture illustrated in Figure 2 can represent arbitrary sentences, including sentences with hierarchical relations as given by relative clauses. Thus, the architecture solves the massiveness of binding problem. It also solves the variable binding problem, because a sentence structure is formed by binding syntax assemblies to each other, in agreement with the sentence structure (as illustrated in Figure 2). Specific words (neuronal word assemblies) are then bound to the variable slots given by the syntax assemblies. For more details and illustrations of the way in which the architecture solves the problems described by Jackendoff (2002), see van der Velde and de Kamps (2006) and van der Velde and de Kamps (2010).

#### 4.4 Multiple architectures interconnected by situ representations

In situ conceptual representations need architectures as illustrated in Figure 2 to combine them in compositional structures. We could assume that each specific kind of compositional structure depends on a specific architecture or ‘neural blackboard’. So, next to the neural blackboard architecture for sentence structure illustrated in Figure 2, there could be neural blackboards for compositional structures like phonological structures, (non-linguistic) sequences of (pseudo)words (Hadley (2008)), compositional structures used in reasoning or complex action sequences.

The use of different blackboards is a direct consequence of the in situ grounded nature of representations. They have to be connected to architectures like blackboards to form compositional structures and to execute processes on the basis of these structures. As Figure 1 suggests, specific blackboards will arise for specific kinds of processes, instead of some general kind of blackboard that could serve for all purposes. When in situ representations are parts of (constituents in) compositional structures, they provide both local and global information. As constituents, they can affect specific (local) compositional structures. But as grounded in situ representations, they

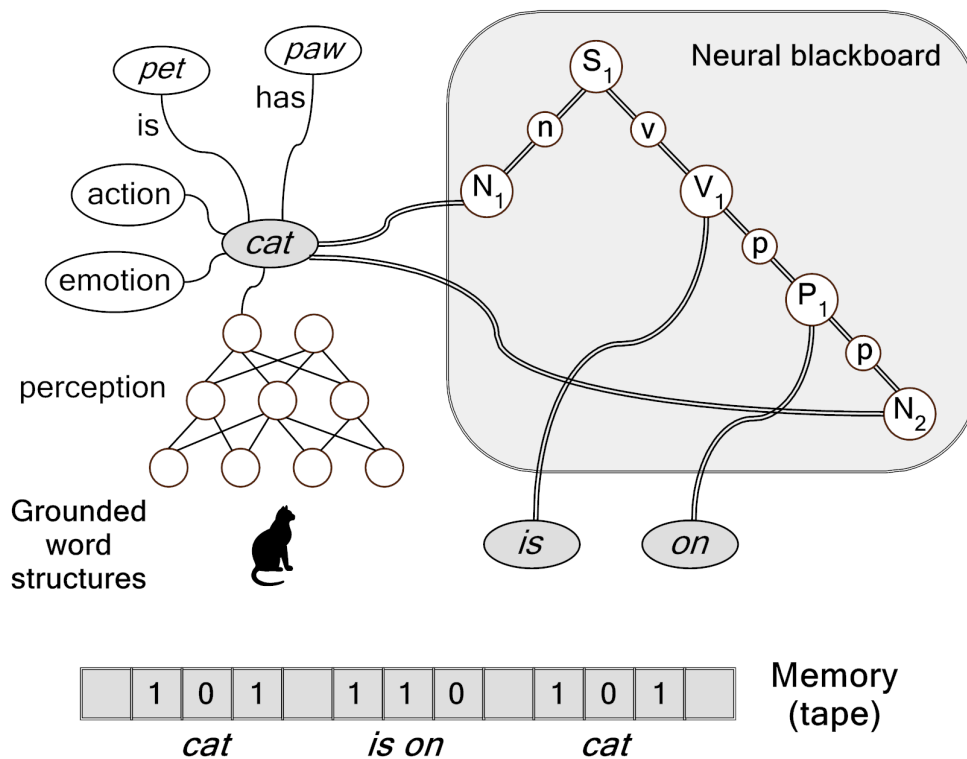


Figure 3: The combinatorial structure *(The) cat is on (the) cat* with grounded representations for the words. Bottom: symbolic representation of *cat is on cat*, with symbol *101* for *cat* and *110* for *is on*.

retain their embedding in the global information structure of which they are a part (van der Velde and de Kamps (2011)). This combination of local and global information is fundamentally lacking in symbolic architectures of cognition (Fodor (2000)). As illustrated in Figure 1, the in situ grounded representations (neuronal assemblies) are part of each specific blackboard architecture in which they are embedded. In this way, they form a link between these architectures. When a process occurs in one architecture, the in situ representations it activates can also induce processes in other architectures, which could in turn influence the process in the first architecture. In this way, an interaction occurs between local information embodied in specific blackboards and global information embodied in in situ grounded representations.

In situ representations are also ideally suited for learning constructions in a more autonomous way. In the next section I will outline a research programme for learning constructions in the architecture described above.

## 5. Autonomous development with in situ representations

As noted, in situ representations as illustrated in Figure 1 share some characteristics with assemblies proposed by Hebb (1949). One of these is their ability to continuously develop and adjust over time. This ability derives from their in situ character. In symbolic architectures, a token of a symbol can

be activated at a given time without activating other tokens of the same symbol. But when a concept is represented by an in situ representation such as a neuronal assembly, the ‘same’ assembly is (partly) activated whenever the concept is active in a cognitive sense. This allows newly processed information to integrate with the neuronal assembly derived from all previous experience related to the concept. This integration is automatic, based on the reactivation of the in situ representation at hand and the context in which it appears. There is no need to develop a program that controls whether a given symbol copy is used in a potentially new context, so that the information related to the symbol needs to be updated.

### 5.1 Learning constructions with in situ representations

A similar learning process can occur with compositional structures based on neuronal assemblies (in situ representations) in a neural blackboard architecture as illustrated in Figure 2. Initially, the compositional structures in the architecture are bound with sustained (working memory) neuronal activity. Hence, they are lost when the sustained activity disappears. But over time, compositional structures can be bound with synaptic modification as well, resulting in more permanent structures stored in long term memory (van der Velde and de Kamps (2006)).

The role of in situ representations (neuronal assemblies) is crucial in this respect. With each representation of the same compositional structure (proposition), the same in situ word representations are involved. This allows a gradual synaptic structure (neuronal proposition) to be formed between these in situ word representations. Thus, the use of in situ word representations directly aligns and compares each newly represented proposition with the previously represented propositions (stored partly or entirely in long term memory). Alignment of compositional representations forms the basis for learning constructions in language. In situ neuronal assemblies are ideally suited for forming constructions with collocations and distributed equivalences. Consider again a robot that has to operate in an unfamiliar environment. Examples of a construction that a robot could detect in an environment are objects that are placed on a desk or table. A construction could be *X on desk*. This construction can be learned when a set of propositions of this kind is stored in long term memory.

Due to the in situ nature of representations, all propositions of the kind *X on desk* consist of the same in situ representations of *on* and *desk*. Thus, with in situ grounded representations (neuronal assemblies), the construction on desk directly results from the collocations between these propositions. In fact, once the construction *on desk* is beginning to develop in long term memory, any proposition represented in the proposition/sentence architecture strengthens the *on desk* structure in long term memory, due to collocation (given by the renewed activation of the in situ grounded assemblies for on and desk). Only the *X* part of the construction is not stored in this way, because it consists of a set of varying objects, each one presented only once or twice. However, when these objects possess a common characteristic shared in their neuronal assemblies (e.g., a specific common visual feature), that characteristic (assembly part) will be connected as the *X* part to the construction *X on desk*. The common characteristic identifies the distributed equivalence that can fill the *X* slot in the construction. Any novel object with the same characteristic (the common visual feature) can fill the *X* slot as well, without any learning required. This aspect of constructions makes language productive.

Learning the construction *X on desk* depends on the ability to represent propositions in an architecture as illustrated in Figure 2. This architecture could be preprogrammed as a beginning

architecture. Control networks have been trained for this architecture, using a basic set of sentences. With these control networks, the architecture can process a substantial set of natural language sentences (van der Velde and de Kamps (2010)).

But we could also start with a more basic architecture, which develops constructions based on sequences of objects or events. These constructions could form the basis for a linguistic architecture, or for learning structural relations between the objects in the environment. Learning these constructions begins when sequences of objects are aligned based on the objects they have in common. In particular, when two or more specific objects in the same sequential order occur in two sequences, these sequences can be aligned, using the matching objects as an anchor. This requires an architecture that forms alignments of sequences based on the in situ grounded representations of the objects in the sequences. Because the representations are in situ, the same representations of matching objects occur in each sequence. The sequences can be embedded in a neural blackboard for sequence representation (van der Velde and de Kamps (2011), which is a more basic architecture compared to the neural blackboard for sentence representation illustrated in Figure 2. In the sequence blackboard structure assemblies for sequential order (first, second, etc.) bind to in situ representations and to each other, forming a sequential representation.

Alignments between different sequences result from the in situ representations they share. The reactivation of an in situ representation will result in the reactivation of the sequence assemblies to which it is bound. Over time, these sequence representations can merge, in particular when they are part of often repeated (sub) sequences of the same in situ representations. The merged sequence representations provide the basis for both the collocations and the distributed equivalences. The difference between the two depends on the conceptual representations to which the sequence representations are bound.

Consider again the construction *X on desk*. This construction could result from merging a number of sequences of the form *X on desk*. Because the in situ representations of *on* and *desk* are repeated between these sequences, their collocation forms the basis for the construction. For the variable *X* there is no single conceptual representation that binds to the sequence. But the objects that can be used as *X* in *X on desk* could share particular characteristics. For example, they could be relatively small, or they could be used for specific purposes. Such shared characteristics are reflected in the overlap between the conceptual (in situ) representations for these objects. This overlap representation (which is also in situ and thus grounded) could bind to the sequence *X on desk*, forming the distributed equivalence for the construction. The next section discusses pilot computer simulations aimed to show that collocations of this kind can develop on the basis of sequential representations of in situ representations.

## 5.2 A simulation of learning constructions with in situ representations

As illustrated in Figure 1, in situ representations are embedded in several “blackboard” architectures, depending on the nature of the concept involved. Interactions between the architectures proceed through the in situ representations (neuronal assemblies) they share.

Among the “other blackboards” are blackboards for sequential processing. A basic model of such a blackboard is illustrated in figure 4. The model consists of a set of “sequence (S) nodes.” They are sparsely connected with random connections. There are also diffuse (overall) connections between S nodes and in situ concept representations. The onset of representing (and learning) a sequence of in situ concepts begins with the “start” activation of a random subset of S nodes.

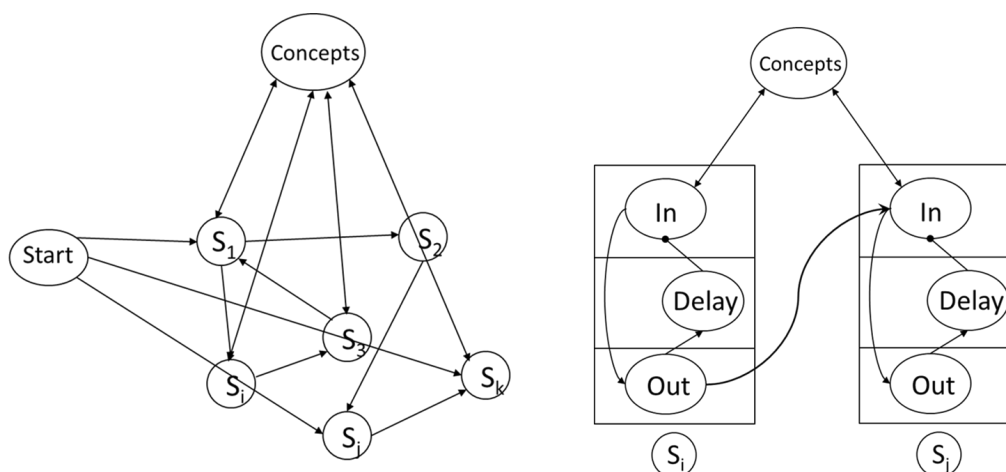


Figure 4: Left: A blackboard architecture for sequential processing, consisting of (more or less) sparsely interconnected sequence nodes. Right: a more detailed illustration of two interconnected sequence nodes.

The sequence (S) nodes have a “columnar structure.” That is, each S node consists of a set of interconnected neuronal populations that form a circuit. The In population is activated by a previously active S node and is interconnected with the in situ concept representations. After being activated, the In population activates the Out population, which can activate the next S node in the sequence. The Out population also activates a Delay population. This population can remain active for a while (based on reverberating activity). It inhibits the In population, which prevents reactivation of S node during the ongoing sequence representation.

The dynamics of the blackboard is as yet simple, based on activation rules derived from connectionism (e.g., Bechtel and Abrahamsen (2002)). An S node is activated by a “previous” (associated) S node and by the associated concept. At each step, a set of S nodes will be activated based on the connection structure in the overall network. However, a winner take all process selects the most active S node at each moment. When an S node is selected, Hebbian learning enhances the connection between the In population of the selected S nodes and the Out population of the previously activated S node. This enhances the connection structure of a sequence in the model. The connection between the In population of the selected S node and currently active concept representation is also enhanced by Hebbian learning. In this way, a sequence of concept representations can be formed, with concepts associated with successively active S nodes.

Figure 5 illustrates how the dynamics in a simulation of the model selects a sequence of S nodes and associated concepts (one for each S node). Even with sparse random S node connections (10-30%) sequences are possible up to number of S nodes. In figure 5 (left) the sequence  $C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow C_4 \rightarrow \text{End}$  is represented. Each concept  $C_i$  is associated with an (initially) arbitrary S node. The connections between the S nodes depend on the random connection structure in the model. It is assumed that each end of a sequence of concepts is marked by a signal (End). This signal is also associated with an S node and marks the end of the sequence.

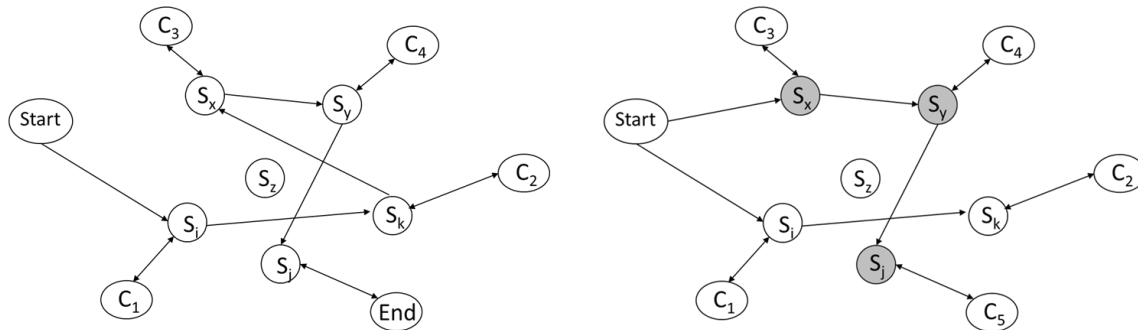


Figure 5: Sequence representation of concepts in the model illustrated in figure 4. Left: the sequence  $C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow C_4 \rightarrow \text{End}$ . Right: the sequences  $C_1 \rightarrow C_2$  and  $C_3 \rightarrow C_4 \rightarrow C_5$ .

The use of the Delay population in the S nodes is crucial for the representation of a sequence. Without it, an S node can easily be connected to more than one other S node (depending on the random activations and current activations in the model). In that case,  $S_i$  could be associated with  $S_j$ , which in turn could be associated with  $S_i$ , which disrupts the sequential order in the representation of the sequence of concepts. With the use of the Delay population in the S nodes, the model can also represent multiple sequences simultaneously, as illustrated with the sequences  $C_1 \rightarrow C_2$  and  $C_3 \rightarrow C_4 \rightarrow C_5$  in figure 5 (here and further, the representation of End is ignored).

When a sequence has been learned by the model, the model can detect the similarity with a new sequence presented to it. In this case, the Delay populations are turned off. So, when the sequence  $C_1 \rightarrow C_2$  has been learned, it can be presented again. It will (most likely) reactivate the same S nodes due to the Hebbian learning between concepts and S nodes and between the S nodes themselves. In this way, the connection structure of the sequence  $C_1 \rightarrow C_2$  can be further enhanced in the model.

But the model can also separately represent different sequences with similar concepts, as illustrated with the sequences  $C_1 \rightarrow C_2$  and  $C_1 \rightarrow C_2 \rightarrow C_3$  in figure 6. This is possible by using a “rerun” option. When the sequence  $C_1 \rightarrow C_2$  is represented, the sequence  $C_1 \rightarrow C_2 \rightarrow C_3$  will activate the same S nodes for  $C_1$  and  $C_2$ , because they are the same in situ representations in both cases. But the activation of  $C_3$  can be used to detect that the two sequences do not match. This can be used to initiate a rerun option for representing the sequence  $C_1 \rightarrow C_2 \rightarrow C_3$ . In that case, the sequence is represented again to the model, but with the Delay populations activated by  $C_1 \rightarrow C_2$  still active. These populations prevent the reactivation of the sequence  $C_1 \rightarrow C_2$ , which allows the formation of a different sequence for  $C_1 \rightarrow C_2 \rightarrow C_3$ .

Figure 7 illustrates how collocations can develop, based on in situ concept representations. Here, the concepts  $C_1$ ,  $C_2$  and  $C_3$  are “words”, consisting of nouns ( $C_1$ , and  $C_3$ ) and a verb ( $C_2$ ). The in situ representations of nouns have an overlap in their representation that represents the

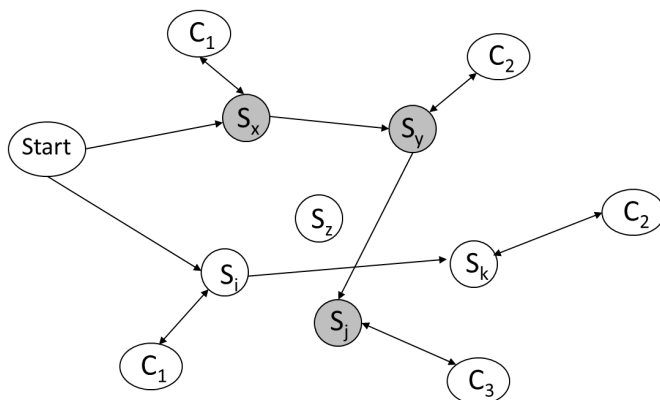


Figure 6: Sequence representation of multiple sequences with similar concepts, here sequences  $C_1 \rightarrow C_2$  and  $C_1 \rightarrow C_2 \rightarrow C_3$ .

“word feature” noun (N). This part of their representations can also be associated to the sequence nodes. This could perhaps occur in a different higher-level sequence blackboard, used for detecting collocations. This higher-level sequence blackboard interacts with the blackboard used for basic sequence representations. As figure 7 illustrates, word features indeed become associated with the S nodes in the model. When the associations between word features and the S nodes in this sequence are further enhanced, the word features can also guide the representation of a N-V-N sequence based on the different words, say  $C_4, C_5$  and  $C_6$ . That is, the sequence  $C_4 \rightarrow C_5 \rightarrow C_6$  is represented by the same S nodes, because  $C_4$  and  $C_6$  are nouns, and  $C_5$  is a verb. In this way, the sequence N-V-N is learned by the model, which represents a basic construction. The individual words associated with the word features become the distributed equivalences for this construction.

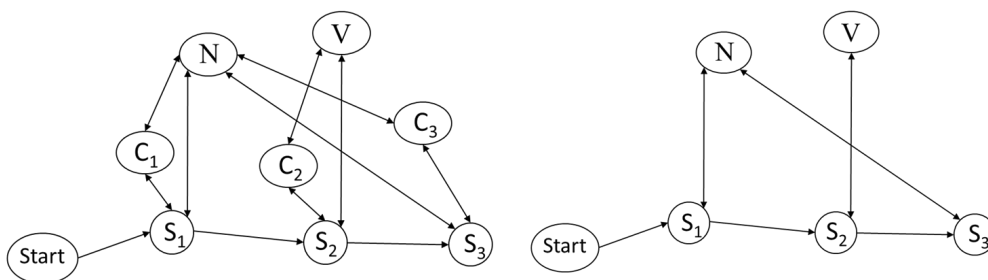


Figure 7: Representation of collocations, based on overlapping representations between in situ concepts. Left: “words” ( $C_1, C_2, C_3$ ) and “word features” (N = noun, V = Verb). Right: Word features only.

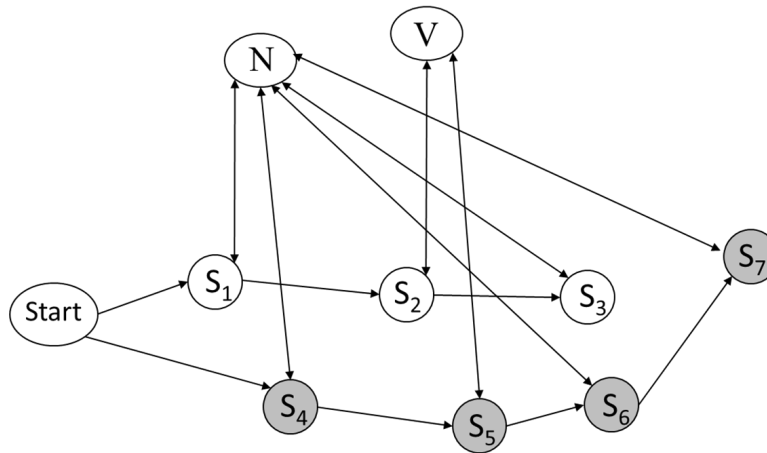


Figure 8: Multiple representations of word feature sequences (constructions). Here,  $N \rightarrow V \rightarrow N$  and  $N \rightarrow V \rightarrow N \rightarrow N$  ( $N$  = noun,  $V$  = Verb).

Finally, figure 8 illustrates that the model can learn to separately represent different constructions that share word features, as  $N$ - $V$ - $N$  and  $N$ - $V$ - $N$ - $N$  (the *give* construction) in this case. The process is based on the rerun process illustrated in figure 6. In the rerun process, the basic blackboard for sequence representation (illustrated in figure 5) is first used to represent a sequence, say a word sequence of the form  $N$ - $V$ - $N$ - $N$ . Then the sequence is presented to a higher level sequence blackboard, as illustrated in figure 7. When a mismatch is detected in this blackboard (between  $N$ - $V$ - $N$  and  $N$ - $V$ - $N$ - $N$ ), the sequence from the basic blackboard can be reactivated and presented again from the start to the higher level blackboard, in which the Delay populations are still active. This results in a new (separate) sequence representation for the sequence  $N$ - $V$ - $N$ - $N$  word sequence in the higher level blackboard. This eventually results in the development of the construction  $N$ - $V$ - $N$ - $N$  in that blackboard. The use of in situ representations is crucial in this process, because they integrate the different blackboards, initiate the matching process and ensure the development of collocations and distributed equivalences.

## 6. Discussion

Clearly, the process outlined above is just a first step towards the development of a full self-organizing architecture for artificial general intelligence. But the use of in situ grounded representations may be an important aspect of this development. For example, in situ grounded representations provide direct information about whether two (or more) objects or events encountered in different sequences are the same or different. When they are the same, they activate the same in situ representation. So, when they are different, they activate different representations. That is, in the case of different objects or events, the activated representation of the object at a given location in the second sequence is not a reactivation of the representation of the object at that location in the first sequence. This lack of reactivation can be represented in the architecture by creating an ‘open slot’ or ‘variable binding site’ in the representation of the collocation generated by the remaining

parts of the sequence. At the same time, the object or event that can fill this variable binding site can be bound to a tag that represents its use as a specific variable.

The representations of constructions based on collocations and the variable slots they contain could further develop in other more hierarchical neural blackboards. Again, the link between these different kinds of blackboards is given by the in situ representations they share. And the forming of higher level constructions would again result from the processes of alignment, collocations and distributed equivalences, taking the constructions and compositional structures of the more basic neural blackboards as input. For example, in the development of a propositional/sentence blackboard, more abstract higher order representations (e.g., ‘noun’, ‘verb’) can be bound to the concepts in the constructions developed in the sequence blackboard outlined in the previous section. For example, with the construction *X on desk*, the propositional/sentence blackboard could relate on desk to other prepositional phrases, forming the basis of a preposition construction, in line with the hierarchical relations between constructions found in natural language.

### 6.1 In situ self-organizing architecture for artificial general intelligence

To what extent can an architecture as presented here constitute a self-organizing architecture for artificial general intelligence? The answer depends, in part, on what self-organizing refers to. If it refers to an architecture that can (re)write its own software, this architecture does not fulfill that goal.

But it is questionable whether artificial general intelligence can be achieved with architectures that can (re)write their software. Consider, for example, IBM’s Watson computer that won the game of Jeopardy. Suppose that its program was based on self-organization, which gradually rewrote the software to the level of winning Jeopardy. This would have been a marvelous achievement, with a lot of potential for applications. But it also raises important questions concerning the future of artificial general intelligence. Watson is a supercomputer, almost needing a dedicated power plant for its energy supply. It is hard to see how such a ‘heavy’ combination could ever result in an autonomous agent that could operate in arbitrary environments, developing artificial general intelligence by its interactions with the world. Perhaps it could rely on a remotely controllable body (avatar). But this would rule out operating in remote environments, due to the disruptive effects of signal transmission delays.

Another option would be the reduction of the hardware and energy consumption, allowing the same self-organizing architecture to be implemented in a mobile and energy reduced manner. However, the chances of this happening are slim. The reduction of ‘traditional’ computer hardware is reaching its limits, and with it the potential for such a reduction of a self-organizing architecture capable of (re)writing its software.

New parallel operating forms of hardware do emerge, such as graphics processing units (GPUs). But it turns out that it is very hard to program these architectures for general purposes, in such a way that they optimally utilize their parallelism. In fact, it is very hard to develop computer languages (e.g., parallel versions of C or C++) that can be used to write these programs. Without such languages, it will be very hard to design self-organizing architectures that can (re)write their software on parallel operating hardware.

Perhaps, therefore, we should also consider other options. One option might be architectures in which hardware and functionality co-evolve. For example, figures 7 and 8 illustrate how a higher level sequential blackboard could develop constructions based on sequential representations in a

lower level sequence blackboard. One could perhaps implement this process on the basis of a recruitment of additional (parallel operating) layers of hardware. Each layer self-organizes based on its dynamics, initial structure and information it is given (and can process). Once organized, it can be the basis for a higher layer of hardware that can begin to self-organize on the basis of the information proved by the lower layer. Gradually new layers can emerge in this way, each processing information in an increasingly more abstract way.

However, it is unclear at present how many hierarchical layers of construction forming would be needed to arrive at constructions that would resemble aspects of self-reflection. A further issue is the reliance on neuronal structures and dynamics in the development of constructions in the architecture outlined here. Neuronal assemblies as illustrated in Figure 1 are in situ representations by default. This may be more than a coincidence. The combination of grounding and cognitive productivity is perhaps a unique feature of the human brain, and in situ representations are crucial in this combination. The question remains whether this entails that we need the development of more brain-like parallel forms of hardware to develop a self-organizing architecture for artificial general intelligence, with an autonomy comparable to that of the human mind/brain, or whether we could emulate this kind of architecture with more digital forms of (parallel) hardware. The answer to this question will perhaps co-develop with the development of such an autonomous architecture itself.

## References

- Anderson, J. R. 1983. *The architecture of cognition*. Cambridge, MA: Harvard University Press.
- Barlow, H. B. 1990. Conditions for versatile learning, Helmholtz's unconscious inference, and the task of perception. *Vision Research*, 30: 1561-1571.
- Barsalou, L. W. 1999. Perceptual symbol systems. *Behavioural and Brain Sciences*, 22: 577-660.
- Bechtel, W., and Abrahamsen, A. 2002. *Connectionism and the mind: Parallel processing, dynamics, and evolution in networks*. Second Edition. Oxford: Basil Blackwell.
- Culicover, P. W., and Jackendoff, R. 2005. *Simpler syntax*. Oxford: Oxford University Press.
- Edelman, S. 2008. *Computing the mind*. Oxford: Oxford University Press.
- Fillmore, C. J. 1965. *Indirect object constructions and the ordering of transformations*. The Hague: Mouton.
- Fodor, J. A., and Pylyshyn, Z. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*. 28: 371.
- Fodor, J. A. 2000. *The mind doesn't work that way*. Cambridge, MA: MIT Press.
- Goldberg, A. E. 1995. *Constructions*. Chicago: University of Chicago Press.
- Goldberg, A. E. 2006. *Constructions at work*. Oxford: Oxford University Press.
- Hadley, R. 2008. The problem of rapid variable creation. *Neural Computation*, 20: 123.

- Harnad, S. 1991. The symbol grounding problem. In S. Forrest (ed.) *Emergent computation: Self-organizing, collective, and cooperative phenomena in natural and artificial computing networks*. Cambridge, (pp. 335-346). MA: MIT Press.
- Harris, Z. S. 1968. *Mathematical structures of language*. New York: Wiley.
- Hebb, D. O. 1949. *The Organization of Behavior*. New York: Wiley.
- Jackendoff, R., and Pinker, S. 2005. The nature of the language faculty and its implications for the evolution of language. *Cognition*, 97: 211-225.
- Jackendoff, R. 2002. *Foundations of language*. Oxford: Oxford University Press.
- Lakoff, G. 1987. *Women, fire, and dangerous things: What categories reveal about the mind*. Chicago: University of Chicago Press.
- Marcus, G. F. 2001. *The algebraic mind*. Cambridge, MA: MIT Press.
- Mojsilovic, A. 2005. A computational model for color naming and describing color composition of images. *IEEE Transactions on Image processing*, 14: 690-699.
- Newell, A. 1990. *Unified theories of cognition*. Cambridge, MA: Harvard University Press.
- Piaget, J. 1950. *The Psychology of Intelligence*. New York: Routledge.
- Pinker, S. 1998. *How the mind works*. London: Penguin.
- Roy, D. 2005. Grounding words in perception and action: computational insights. *Trends in Cognitive Sciences*, 9: 389-395.
- Thompson, P. W. 1995. Constructivism, cybernetics, and information processing: Implications for research on mathematical learning. In L. P. Streffe and J. Gale (Eds.), *Constructivism in education* (pp. 123-14). Hillsdale, NJ: Erlbaum.
- Thórisson, K. R. 2009. From constructionist to constructivist A.I. *Keynote, AAAI Fall Symposium-Biologically Inspired Cognitive Architectures*, 175-183. AAI Tech Report FS-09-01, AAAI press, Menlo Park, CA.
- van der Velde, F., and de Kamps, M. 2006. Neural blackboard architectures of combinatorial structures in cognition. *Behavioral and Brain Sciences*, 29: 37-70.
- van der Velde, F., and de Kamps, M. 2010. Learning of control in a neural architecture of grounded language processing. *Cognitive Systems Research*, 11: 93107.
- van der Velde, F., and de Kamps, M. 2011. Compositional connectionist structures based on in situ grounded representations. *Connection Science*, 23: 97-107.
- van der Velde, F. 2010. Where Artificial Intelligence and Neuroscience Meet: The Search for Grounded Architectures of Cognition. *Advances in Artificial Intelligence*, 2010: 1-18. doi:10.1155/2010/918062.