

**Neural Blackboard Architectures
of Combinatorial Structures in Cognition**

Frank van der Velde

Unit of Cognitive Psychology

Leiden University

Wassenaarseweg 52, 2333 AK Leiden

The Netherlands

vdvelde@fsw.leidenuniv.nl

Abstract

Human cognition is unique in the way in which it relies on combinatorial (or compositional) structures. Language provides ample evidence for the existence of combinatorial structures, but they can also be found in visual cognition. To understand the neural basis of human cognition, it is therefore essential to understand how combinatorial structures can be instantiated in neural terms. In his recent book on the foundations of language, Jackendoff formulated four fundamental problems for a neural instantiation of combinatorial structures: the massiveness of the binding problem, the problem of 2, the problem of variables and the transformation of combinatorial structures from working memory to long-term memory. This paper aims to show that these problems can be solved by means of neural 'blackboard' architectures. For this purpose, a neural blackboard architecture for sentence structure is presented. In this architecture, neural structures that encode for words are temporarily bound in a manner that preserves the structure of the sentence. It is shown that the architecture solves the four problems presented by Jackendoff. The ability of the architecture to instantiate sentence structures is illustrated with examples of sentence complexity observed in human language performance. Similarities exist between the architecture for sentence structure and blackboard architectures for combinatorial structures in visual cognition, derived from the structure of the visual cortex. These architectures are briefly discussed, together with an example of a combinatorial structure in which the blackboard architectures for language and vision are combined. In this way, the architecture for language is grounded in perception.

Content

1. Introduction
 2. Four challenges for cognitive neuroscience
 - 2.1. The massiveness of the binding problem
 - 2.2. The problem of 2
 - 2.2.1. The problem of 2 and the symbol grounding problem
 - 2.3. The problem of variables
 - 2.4. Binding in working memory versus long-term memory
 - 2.5. Overview
 3. Combinatorial structures with synchrony of activation
 - 3.1. Nested structures with synchrony of activation
 - 3.2. Productivity with synchrony of activation
 4. Processing linguistic structures with recurrent neural networks
 - 4.1. Combinatorial productivity with RNNs
 - 4.2. RNNs and the massiveness of the binding problem
 5. Blackboard architectures of combinatorial structures
 6. A neural blackboard architecture of sentence structure
 - 6.1. Gating and memory circuits
 - 6.2. Overview of the architecture
 - 6.2.1. Connection structure for binding in the architecture
 - 6.3. Multiple instantiation and binding in the architecture
 - 6.3.1. Answering binding questions
 - 6.4. Extending the blackboard architecture
 - 6.4.1. The modular nature of the blackboard architecture
 - 6.5. Constituent binding in long-term memory
 - 6.5.1. One-trial learning
 - 6.5.2. Explicit encoding of sentence structure with synaptic modification
 - 6.6. Variable binding
 - 6.6.1. Neural structure versus spreading of activation
 - 6.7. Structural dependencies in the blackboard architecture
 - 6.7.1. Embedded clauses in the blackboard architecture
 - 6.7.2. Multiple embedded clauses
 - 6.7.3. Dynamics of binding in the blackboard architecture
 - 6.7.4. Dynamics of binding and complexity
 - 6.8. Further development of the architecture
 7. Neural blackboard architectures of combinatorial structures in vision
 - 7.1. Feature binding
 - 7.2. A neural blackboard architecture of visual working memory
 - 7.2.1. Feature binding in visual working memory
 - 7.3. Feature binding in long-term memory
 - 7.4. Integrating combinatorial structures in language and vision
 8. Conclusion
- Notes
- References

1. Introduction

Human cognition is unique in the manner in which it processes and produces complex combinatorial (or compositional) structures (e.g., Anderson 1983; Newell 1990; Pinker 1998). Therefore, to understand the neural basis of human cognition, it is essential to understand how combinatorial structures can be instantiated in neural terms. However, combinatorial structures present particular challenges to theories of neurocognition, which have not been widely recognized in the cognitive neuroscience community (Jackendoff 2002).

A prominent example of these challenges is given by the neural instantiation (in theoretical terms) of linguistic structures. In his recent book on the foundations of language, Jackendoff (2002; see also Jackendoff in press) analyzed the most important theoretical problems that the combinatorial and rule-based nature of language presents to theories of neurocognition. He summarized these problems under the heading of ‘four challenges for cognitive neuroscience’ (pp. 58-67). As recognized by Jackendoff, these problems arise not only with linguistic structures, but with combinatorial cognitive structures in general.

This paper aims to show that neural ‘blackboard’ architectures can provide an adequate theoretical basis for a neural instantiation of combinatorial cognitive structures. In particular, I will discuss how the problems presented by Jackendoff (2002) can be solved in terms of a neural blackboard architecture of sentence structure. I will also discuss the similarities between the neural blackboard architecture of sentence structure and neural blackboard architectures of combinatorial structures in visual cognition and visual working memory (Van der Velde 1997; Van der Velde & de Kamps 2001; 2003a).

To begin with, I will first outline the problems described by Jackendoff (2002) in more detail. This presentation is followed by a discussion of the most important solutions that have been offered thus far to meet some of these challenges. These solutions are based on either synchrony of activation or on recurrent neural networks¹.

2. Four challenges for cognitive neuroscience

The four challenges for cognitive neuroscience presented by Jackendoff (2002) consists of: the massiveness of the binding problem that occurs in language, the problem of multiple instances (or the ‘problem of 2’), the problem of variables, and the relation between binding in working memory and binding in long-term memory. I will discuss these problems in turn.

2.1. The massiveness of the binding problem

In neuroscience, the binding problem concerns the way in which neural instantiations of elements (constituents) can be related (bound) temporarily in a manner that preserves the structural relations between the constituents. Examples of this problem can be found in visual perception. Colors and shapes of objects are partly processed in different brain areas, but we perceive objects as a unity of color and shape. Thus, in a visual scene with a green apple and a red orange, the neurons that code for green have to be related (temporarily) with the neurons that code for apple, so that the confusion with a red apple (and a green orange) can be avoided.

In the case of language, the problem is illustrated in figure 1. Assume that words like *cat*, *chases* and *mouse* each activate specific neural structures, such as the ‘word

assemblies' discussed by Pulvermüller (1999). The problem is how the neural structures or word assemblies for *cat* and *mouse* can be bound to the neural structure or word assembly of the verb *chases*, in line with the thematic roles (or argument structure) of the verb. That is, how *cat* and *mouse* can be bound to the role of agent and theme of *chases* in the sentence *The cat chases the mouse*, and to the role of theme and agent of *chases* in the sentence *The mouse chases the cat*.

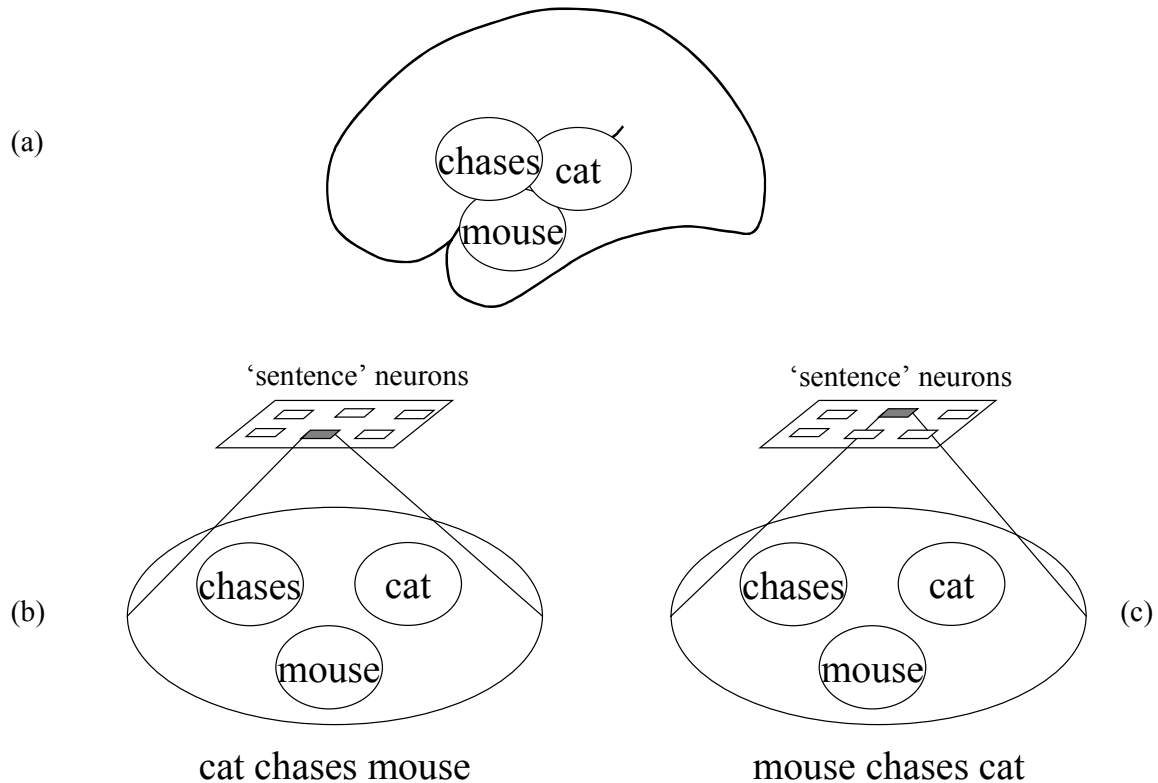


Figure 1. (a). Illustration of the neural structures ('neural word assemblies') activated by the words *cat*, *chases* and *mouse*. Bottom: An attempt to encode sentence structures with specialized 'sentence' neurons. In (b), a 'sentence' neuron has the assemblies for the words *cat*, *chases* and *mouse* in its 'receptive field' (as indicated with the cone). The neuron is activated by a specialized neural circuit when the assemblies in its receptive field are active in the order *cat chases mouse*. In (c), a similar 'sentence' neuron for the sentence *mouse chases cat*.

A potential solution for this problem is illustrated in figure 1. It consists of specialized neurons (or populations of neurons) that are activated when the strings *cat chases mouse* (figure 1b) or *mouse chases cat* (figure 1c) are heard or seen. Each neuron has the word assemblies for *cat*, *mouse* and *chases* in its 'receptive field' (illustrated with the cones in figures 1b and 1c). Specialized neural circuits could activate one neuron in the case of *cat chases mouse* and another neuron in the case of *mouse chases cat*, by using the difference in temporal word order in both strings. Circuits of this kind can be found in the case of motion detection in visual perception (e.g., Hubel 1995). For

instance, the movement of a vertical bar that sweeps across the retina in the direction from A to B can be detected by using the difference in activation time (onset latency) between the ganglion cells in A and B. A similar specialized circuit can detect a vertical bar moving from B to A.

However, a fundamental problem with this solution in the case of language is its lack of productivity. Only specific and familiar sentences can be detected in this way. But any novel sentence of the type *Noun chases Noun* or, more generally, *Noun Verb Noun* will not be detected because the specific circuit (and neuron) for that sentence will be missing. Yet, when we learn that *Dumbledore is headmaster of Hogwarts*, we immediately understand the meaning of *Dumbledore chases the mouse*, even though we have never encountered that sentence before.

The difference between language and motion detection in this respect illustrates a fundamental difference in nature between these two cognitive processes. In the case of motion detection there is a limited set of possibilities, so that it is possible (and it pays off) to have specialized neurons and neural circuits for each of these possibilities. However, this solution is not feasible in the case of language. Linguists typically describe language in terms of its unlimited combinatorial productivity. Words can be combined into phrases, which in turn can be combined into sentences, so that arbitrary sentence structures can be filled with arbitrary arguments (e.g., Weibelhuth 1995; Sag & Wasow 1999; Chomsky 2000; Pullum & Scholz 2001; Jackendoff 2002; Piattelli-Palmarini 2002). In theory, an unlimited amount of sentences can be produced in this way, which excludes the possibility of having specialized neurons and circuits for each of these sentences.

One could argue that many of the sentences that are theoretically possible may be too complex for humans to understand (Christiansen & Chater 1999). However, unlimited (recursive) productivity is not necessary to make a case for the combinatorial nature of language, given the number of sentences that can be produced or understood. For instance, the average English-speaking 17-year-old knows more than 60,000 words (Bloom 2000). With this lexicon, and with a limited sentence length of 20 words or less, one can produce a set of sentences in natural language in the order of 10^{20} or more (Pinker 1998). A set of this kind can be characterized as a 'performance set' of natural language, in the sense that (barring a few selected examples) any sentence from this set can be produced or understood by a normal language user. Such a performance set is not unlimited, but it is of 'astronomical' magnitude (e.g., 10^{20} exceeds the estimated lifetime of the universe expressed in seconds). By consequence, most sentences in this set are sentences that we have never heard or seen before. Yet, because of the combinatorial nature of language we have the ability to produce or understand arbitrary sentences from a set of this kind.

Hence, the set of possibilities that we can encounter in the case of language is unlimited in any practical sense. This precludes a solution of the binding problem in language in terms of specialized neurons and circuits. Instead, a solution is needed that depends on the ability to bind arbitrary arguments to the thematic roles of arbitrary verbs, in agreement with the structural relations expressed in the sentence. Moreover, the solution has to satisfy the massiveness of the binding problem as it occurs in language, which is due to the often complex and hierarchical nature of linguistic structures. For instance, in the sentence *The cat that the dog bites chases the mouse*, *cat* is bound to the

role of theme of the verb *bites*, but it is bound to the role of agent of the verb *chases*. In fact, the whole phrase *The cat that the dog bites* is bound to the role of agent of the verb *chases* (with *cat* as the head of the phrase). Each of these specific bindings has to be satisfied in an encoding of this sentence. Further examples can be seen in a simple syntactic structure like *beside a big star* (Jackendoff 2002). Here, one can identify relationships like ‘it is a prepositional phrase’, ‘it is a part of a verb phrase’, ‘it follows a verb’, and ‘it has a preposition and noun phrase parts’. Binding problems occur for each of these relationships.

2.2. The problem of 2

The second problem presented by Jackendoff (2002) is the problem of multiple instances, or the ‘problem of 2’. Jackendoff illustrates this problem with the sentence *The little star is beside a big star*². The word *star* occurs twice in this sentence, the first time related with the word *little* and the second time related with the word *big*. The problem is how in neural terms the two occurrences of the word *star* can be distinguished, so that *star* is first bound with *little* and then with *big*, without creating the erroneous binding of *little big star*. The problem of 2 results from the assumption that any occurrence of a given word will result in the activation of the same neural structure (e.g., its word assembly, as illustrated in figure 1). But if the second occurrence of a word only results in the reactivation of a neural structure that was already activated by the first occurrence of that word, the two occurrences of the same word are indistinguishable (Van der Velde 1999).

Perhaps the problem could be solved by assuming that there are multiple neural structures that encode for a single word. The word *star* could then activate one neural structure in *little star* and a different one in *big star*, so that the bindings *little star* and *big star* can be encoded without creating *little big star*. However, this solution would entail that there are multiple neural structures for all words in the lexicon, perhaps even for all potential positions a word could have in a sentence (Jackendoff 2002).

More importantly even, this solution disrupts the unity of word encoding as the basis for the meaning of a word. For instance, the relation between the neural structures for *cat* and *mouse* in *cat chases mouse* could develop into the neural basis for the long-term knowledge (‘fact’) that cats chase mice. Similarly, the relation between the neural structures for *cat* and *dog* in *dog bites cat* could form the basis of the fact that dogs fight with cats. But if the neural structure for *cat* (say, *cat*₁) in *cat*₁ *chases mouse* is different from the neural structure for *cat* (say, *cat*₂) in *dog bites cat*₂, then these two facts are about different kinds of animals.

2.2.1. The problem of 2 and the symbol grounding problem

It is interesting to look at the problem of 2 from the perspective of the symbol grounding problem that occurs in cognitive symbol systems. Duplicating symbols is easy in a symbol system. However, in a symbol system, one is faced with the problem that symbols are arbitrary entities (e.g., strings of bits in a computer), which therefore have to be interpreted to provide meaning to the system. That is, symbols have to be ‘grounded’ in perception and action if symbol systems are to be viable models of cognition (Harnad 1991; Barsalou 1999).

Grounding in perception and action can be achieved with neural structures such as the word assemblies illustrated in figure 1. In line with the idea of neural assemblies

proposed by Hebb (1949), Pulvermüller (1999) argued that words activate neural assemblies, distributed over the brain (as illustrated with the assemblies for the words *cat*, *mouse* and *chases* in figure 1). One could imagine that these word assemblies have developed over time by means of a process of association. Each time a word was heard or seen, certain neural circuits would have been activated in the cortex. Over time, these circuits will be associated, which results in an overall cell assembly that reflects the meaning of that word. For instance, assemblies for words with a specific visual content would stretch into the visual cortex, whereas words that describe particular actions (e.g., ‘walking’ vs ‘talking’) would activate assemblies that stretch into specific parts of the motor cortex, as observed by Pulvermüller et al. (2001).

But, as argued above, word assemblies are faced with the problem of 2. Thus, it seems that the problem of 2 and the symbol grounding problem are complementary problems. To provide grounding, the neural structure that encodes for a word is embedded in the overall network structure of the brain. But this makes it difficult to instantiate a duplication of the word, and thus to instantiate even relatively simple combinatorial structures such as *The little star is beside a big star*. Conversely, duplication is easy in symbol systems (e.g., if ‘1101’ = *star*, then one would have *The little 1101 is beside a big 1101*, with *little* and *big* each related to an individual copy of 1101). But symbols can be duplicated easily because they are not embedded in an overall structure that provides the grounding of the symbol³.

2.3. The problem of variables

The knowledge of specific facts can be instantiated on the basis of specialized neural circuits, in line with those illustrated in figure 1. But knowledge of systematic facts, such as the fact that $own(y,z)$ follows from $give(x,y,z)$, cannot be instantiated in this way, that is, in terms of a listing of all specific instances of the relation between the predicates *own* and *give* (e.g., from $give(John, Mary, book)$ it follows that $own(Mary, book)$; from $give(Mary, John, pen)$ it follows that $own(John, pen)$; etc.).

Instead, the derivation that $own(Mary, book)$ follows from $give(John, Mary, book)$ is based on the rule that $own(y,z)$ follows from $give(x,y,z)$, combined with the binding of *Mary* to the variable *y* and *book* to the variable *z*. This raises the question of how rule-based derivation with variable binding can be instantiated in the brain.

The ability of rule-based derivation with variable binding provides the basis for the systematic nature of cognition (Fodor & Pylyshyn 1988). Cognition is systematic in the sense that one can learn from specific examples and apply that knowledge to all examples of the same kind. A child will indeed encounter only specific examples (e.g., that when John gives Mary a book, it follows that Mary owns the book) and yet it will learn that $own(y,z)$ follows from all instances of the kind $give(x,y,z)$. In this way, the child is able to handle novel situations, such as the derivation that $own(Harry, broom)$ follows from $give(Dumbledore, Harry, broom)$.

2.4. Binding in working memory versus long-term memory

Working memory in the brain is generally assumed to consist of a sustained form of activation (e.g. Amit 1995; Fuster 1995). That is, information is stored in working memory as long as the neurons that encode the information remain active. In contrast, long-term memory results from synaptic modification. In this way, the connections

between neurons are modified (e.g., enhanced) so that when some of the neurons are reactivated, they will reactivate the others neurons as well. The neural word assemblies, illustrated in figure 1, are formed by this process.

Both forms of memory are related in the sense that information in one form of memory can be transformed into information in the other form of memory. Information is initially stored in working memory before it is stored in long-term memory. Conversely, information in long-term memory can be reactivated and stored in working memory. This raises the question of how the same combinatorial structure can be instantiated both in terms of neural activation and in terms of synaptic modification, and how these different instantiations can be transformed into one another.

2.5. Overview

It is clear that the four problems presented by Jackendoff (2002) are interrelated. For instance, the problem of 2 also occurs in rule-based derivation with variable binding, the massiveness of the binding problem is found in combinatorial structures stored in working memory and in combinatorial structures stored in long-term memory. Therefore, a solution of these problems has to be an integrated one that solves all four problems simultaneously. In this paper, I will discuss how all four problems can be solved in terms of neural blackboard architectures in which combinatorial structures can be instantiated.

First, however, I will discuss two alternatives for a neural instantiation of combinatorial structures. In the next section I will discuss the use of synchrony of activation as a mechanism for binding constituents in combinatorial structures. In the section after that, I will discuss the view that combinatorial structures can be handled with recurrent neural networks.

3. Combinatorial structures with synchrony of activation

An elaborate example of a neural instantiation of combinatorial structures in which synchrony of activation is used as a binding mechanism is found in the model of reflexive reasoning presented by Shastri and Ajjanagadde (1993). In their model, synchrony of activation is used to show how a known fact such as *John gives Mary a book* can result in an inference such as *Mary owns a book*.

The proposition *John gives Mary a book* is encoded by a ‘fact node’ that detects the respective synchrony of activation between the nodes for *John*, *Mary* and *book*, and the nodes for *giver*, *recipient* and *give-object*. These nodes encode for the thematic roles of the predicate *give(x,y,z)*. In a simplified manner, the reasoning process begins with the query *own(Mary, book)?* (i.e., does Mary own a book?). The query results in the respective synchronous activation of the nodes for *owner* and *own-object* of the predicate *own(y,z)* with the nodes for *Mary* and *book*. In turn, the nodes for *recipient* and *give-object* of the predicate *give(x,y,z)* are activated by the nodes for *owner* and *own-object*, such that *owner* is in synchrony with *recipient* and *own-object* is in synchrony with *give-object*. As a result, the node for *Mary* is in synchrony with the node for *recipient* and the node for *book* is in synchrony with the node for *give-object*. This allows the fact node for *John gives Mary a book* to become active, which produces the affirmative answer to the query.

A first problem with a model of this kind is found in a proposition like *John gives Mary a book and Mary gives John a pen*. With synchrony as a binding mechanism, a

confusion arises in this proposition between *John* and *Mary* in their respective roles of *giver* and *recipient* in this proposition. In effect, the same pattern of activation will be found in the proposition *John gives Mary a pen and Mary gives John a book*. Thus, with synchrony of activation as a binding mechanism, both propositions are indistinguishable. It is not difficult to see the problem of 2 here. *John* and *Mary* occur twice in the proposition, but in different thematic roles. The simultaneous but distinguishable binding of *John* and *Mary* with different thematic roles cannot be achieved with synchrony of activation.

To solve this problem, Shastri and Ajjanagadde allowed for a duplication (or multiplication) of the nodes for the predicates. In this way, the whole proposition *John gives Mary a book and Mary gives John a pen* is partitioned into the two elementary propositions *John gives Mary a book* and *Mary gives John a pen*. To distinguish between the propositions, the nodes for the predicate $give(x,y,z)$ are duplicated. Thus, there are specific nodes for, say, $give_1(x,y,z)$ and $give_2(x,y,z)$, with $give_1(x,y,z)$ activated by *John gives Mary a book* and $give_2(x,y,z)$ activated by *Mary gives John a pen*. Furthermore, for the reasoning process to work, the associations between predicates have to be duplicated as well. Thus, the node for $give_1(x,y,z)$ has to be associated with a node for, say, $own_1(y,z)$ and the node for $give_2(x,y,z)$ has to be associated with a node for $own_2(y,z)$.

This raises the question of how these associations can be formed simultaneously during learning. During its development, a child will learn from specific examples. Thus, it will learn that, when *John gives Mary a book*, it follows that *Mary owns the book*. In this way, the child will form an association between the nodes for, say, $give_1(x,y,z)$ and $own_1(y,z)$. But the association between the node for $give_2(x,y,z)$ and $own_2(y,z)$ would not be formed in this case, because these nodes are not activated with *John gives Mary a book* and *Mary owns the book*. Thus, when the predicate $give(x,y,z)$ is duplicated into $give_1(x,y,z)$ and $give_2(x,y,z)$, the systematicity between *John gives Mary a book* and *Mary gives John a pen* is lost.

3.1. Nested structures with synchrony of activation

The duplication solution discussed above fails with nested (or hierarchical) propositions. For instance, the proposition *Mary knows that John knows Mary* cannot be partitioned into two propositions *Mary knows* and *John knows Mary*, because the entire second proposition is the y argument of $knows(Mary, y)$. Thus, the fact node for *John knows Mary* has to be in synchrony with the node for $know-object$ of the predicate $know(x,y)$. The fact node for *John knows Mary* will be activated because *John* is in synchrony with the node for $knower$ and *Mary* is in synchrony with the node for $know-object$. However, the fact node for *Mary knows Mary* will also be activated in this case, because *Mary* is in synchrony with both $knower$ and $know-object$ in the proposition *Mary knows that John knows Mary*. Thus, the proposition *Mary knows that John knows Mary* cannot be distinguished from the proposition *Mary knows that Mary knows Mary*. Likewise, the proposition *Mary knows that John knows Mary* cannot be distinguished from the propositions *John knows that John knows Mary* and *John knows that Mary knows Mary*, because *John* is in synchrony with $knower$ in each of these propositions.

3.2. Productivity with synchrony of activation

A further problem with the use of synchrony of activation as a binding mechanism is its lack of productivity. The model of Shastri and Ajjanagadde depends on the use of fact nodes, such as the fact node for *John gives Mary a book*, to detect the synchrony of activation between arguments and thematic roles. The use of fact nodes is needed because synchrony of activation has to be detected to process the information that it encodes (Dennett 1991). But fact nodes, and the circuits that activate them, are similar to the specialized neurons and circuits illustrated in figure 1. It is excluded to have such nodes and circuits for all possible verb-argument bindings that can occur in language, in particular for novel instances of verb-argument binding. As a result, synchrony of activation as a binding mechanism fails to provide the productivity given by combinatorial structures.

The problems analyzed here, the inability to solve the problem of 2, the inability to deal with nested structures, and the lack of systematicity and productivity, are also found in other domains in which synchrony of activation is used as a binding mechanism, such as visual cognition (Van der Velde & de Kamps 2002).

4. Processing linguistic structures with recurrent neural networks

The argument that combinatorial structures are needed to obtain productivity in cognition has been questioned (Elman 1991; Churchland 1995, Port & Van Gelder 1995). In this view, productivity in cognition can be obtained in a ‘functional’ manner (‘functional compositionality’, Van Gelder 1990), without using explicit combinatorial structures. The most elaborate approach of this kind is found in the processing of linguistic structures with recurrent neural networks (Elman 1991; Miikkulainen 1996; Christiansen & Chater 2001; Palmer-Brown et al. 2002).

A recurrent neural network (RNN) is a multilayer (usually three-layer) feedforward network, in which the activation pattern in the hidden (middle) layer is copied back to the input layer, where it serves as part of the input to the network in the next learning step. In this way, RNNs are capable of processing and memorizing sequential structures. Elman (1991) used RNNs to predict what kind of word would follow next at a given point in a sentence. For instance, in case of the sentence *Boys who chase boys feed cats*, the network had to predict that after *Boys who chase* a noun would follow, and that after *Boys who chase boys* a plural verb would occur. To perform this task, the network was trained with sentences from a language generated with a small lexicon and a basic phrase grammar. The network succeeded in this task, both for the sentences that were used in the training session and with other sentences from the same language.

A more complex model was presented by Miikkulainen (1996). The model consisted of multiple parts (including a ‘parser’), based on RNNs. The purpose of the model was to assign thematic roles (agent, act, patient) to the words in a clause. The model succeeded in this task, even with embedded clauses (however, clauses were restricted to two or three word clauses, which resulted from the fact that the output layer of the parser had three nodes).

Thus, it seems that RNNs are capable to process linguistic structures in a noncombinatorial manner. However, as Christiansen and Chater (2001) noted, all RNNs model languages derived from small vocabularies (in the order of 10 to 100 words). In contrast, the vocabulary of natural language is huge, which results in an ‘astronomical’

productivity when combined with even limited sentence structures (e.g., sentences with 20 words or less, see section 2.1.). Therefore, I will discuss this form of ‘combinatorial’ productivity in the case of language processing with RNNs in more detail.

4.1. Combinatorial productivity with RNNs

In Elman (1991), the RNN was trained and tested with a language in the order of 10^5 sentences, based on a lexicon of about 20 words. In contrast, the combinatorial productivity of natural language is in the order of 10^{20} sentences or more, based on a lexicon of 10^5 words. A basic aspect of such a combinatorial productivity is the ability to insert words from one familiar sentence context into another. For instance, if one learns that *Dumbledore is headmaster of Hogwarts*, one can also understand *Dumbledore chases the mouse*, or *The dog sees Hogwarts*, even though these specific sentences have not been encountered before. RNNs should have this capability as well, if they are to approach the combinatorial productivity of natural language.

Using the prediction task of Elman (1991), we investigated this question by testing the ability of RNNs to recognize a sentence consisting of a new combination of familiar words in familiar syntactic roles (Van der Velde et al. 2003). In one instance, we used sentences like *dog hears cat*, *boy sees girl*, *dog loves girl* and *boy follows cat* to train the network on the word prediction task. The purpose of the training sentences was to familiarize the RNNs with *dog*, *cat*, *boy* and *girl* as arguments of verbs. Then, a verb like *hears* from *dog hears cat* was inserted into another trained sentence like *boy sees girl* to form the test sentence *boy hears girl*, and the networks were tested on the prediction task for this sentence.

To strengthen the relations between *boy*, *hears* and *girl*, we also included training sentences like *boy who cat hears obeys John* and *girl who dog hears likes Mary*. These sentences introduce *boy* and *hears*, and *girl* and *hears*, in the same sentence context (without using *boy hears* and *hears girl*)⁴. In fact, *girl* is the object of *hears* in *girl who dog hears likes Mary*, as in the test sentence *boy hears girl*.

However, although the RNNs learned the training sentences to perfection, they failed with the test sentences. Despite the ability to process *boy sees girl* and *dog hears cat*, and even *girl who dog hears likes Mary*, they could not process *boy hears girl*. The behavior of the RNNs with the test sentence *boy hears girl* was in fact similar to the behavior in a ‘word salad’ condition, which consisted of random word strings, based on the words used in the training session. Analysis of this ‘word salad’ condition showed that the RNNs predicted the next word on the basis of direct word-word associations, based on all two-word combinations found in the training sentences. The similarity between ‘word salads’ and the test sentence *boy hears girl* suggests that RNNs resort to word-word associations when they have to process novel sentences composed of familiar words in familiar grammatical structures.

The results of these simulations indicate that RNNs do not possess a minimal form of the combinatorial productivity that underlies human language processing. To put this in perspective, it is important to realize that the lack of combinatorial productivity observed in these simulations is not just a negative result, that could have been avoided by using a better learning (training) algorithm. The training sentences were learned to perfection. The best that another algorithm could do is to learn these sentences to the same level of perfection. It is unclear how this could produce a different result on the test sentences.

Furthermore, the crucial issue here is not learning, but the contrast in behavior exhibited by the RNNs in these simulations. The RNNs were able to process (‘understand’) *boy sees girl* and *dog hears cat*, and even *girl who dog hears likes Mary*, but not *boy hears girl*. This contrast in behavior is not found in humans, regardless of the learning procedure used. It is not found in human behavior due to the structure of the human language system. This is what the issue of systematicity is all about: if you understand *boy sees girl*, *dog hears cat* and *girl who dog hears likes Mary*, you cannot but understand *boy hears girl*. Any failure to do so would be regarded as pathological⁵.

4.2. RNNs and the massiveness of the binding problem

The simulations discussed above again show that RNNs are capable of processing learned sentences like *girl who dog hears obeys Mary*, and other complex sentence structures. Thus, even though RNNs fail in terms of combinatorial productivity, they could be used to process sentence structures in abstract terms. That is, they could process a sentence structure in terms of *Nouns (N)* and *Verbs (V)*, such as *N-who-N-V-V-N* in the case of sentences like *girl who dog hears obeys Mary*.

Sentence processing in terms of *N-V* strings can be related with the word assemblies illustrated in figure 1. Words of a similar category, like verbs or nouns, would have a common part in their cell assemblies that reflects that they are verbs or nouns. The RNNs could be trained to process sentences in terms of these common parts, thus in terms of *N-V* strings. However, when used in this way, RNNs can only be a part of a neural model of human language performance. Consider, for instance, the sentences *cat chases mouse* and *mouse chases cat*. Both sentences are *N-V-N* sentences, and thus indistinguishable for these RNNs. Yet, the two sentences convey very different messages, and humans can understand these differences. In particular, they can produce the correct answers to the ‘who does what to whom’ questions for each of these sentences, which cannot be answered on the level of the *N-V-N* structure processed by RNNs.

This raises two important questions for the use of RNNs in this manner. First, how is the difference between *cat chases mouse* and *mouse chases cat* instantiated in neural terms? The lack of combinatorial productivity discussed above shows that this cannot be achieved with RNNs. Second, given a neural instantiation of *cat chases mouse* and *mouse chases cat*, how can the structural *N-V* information processed by the RNNs be related with the specific content of each sentence? This is a ‘binding’ problem, because it requires that, for instance, the first *N* in *N-V-N* is bound to *cat* in the first sentence and to *mouse* in the second sentence.

However, even if these problems are solved, sentence processing in terms of *N-V* strings is still faced with serious difficulties, as illustrated with the following sentences:

- The cat that the dog that the boy likes bites chases the mouse* (1)
The fact that the mouse that the cat chases roars surprises the boy (2)

The abstract (*N-V*) structure of both sentences is the same: *N-that-N-that-N-V-V-N*. Yet, there is a clear difference in complexity between these sentences (Gibson 1998). Sentences with complement clauses (2) are much easier to process than sentences with center-embeddings (1). This difference can be explained in terms of the bindings (dependencies) within the sentence structures. In (1) the first noun is related with the

second verb as its object (theme) and with the third verb as its subject (agent). In (2), the first noun is only related with the third verb (as its subject). This difference in structural dependency (binding) is not captured in the sequence *N-that-N-that-N-V-V-V-N*.

The structural dependencies that constitute the difference between sentences (1) and (2) again illustrate the massiveness of the binding problem that occurs in linguistic structures. Words and clauses have to be bound correctly to other words and clauses in different parts of the sentence, in line with the hierarchical structure of a sentence. These forms of binding are clearly beyond the capacity of language processing with RNNs. Similar limitations are found with RNNs in case of the problem of variables (Marcus 2001).

5. Blackboard architectures of combinatorial structures

A combinatorial structure consists of parts (constituents) and their relations. Briefly stated, one could argue that the lack of combinatorial productivity with RNNs, as discussed above, illustrates a failure to encode the individual parts (words) of a combinatorial structure (sentence) in a productive manner. In contrast, synchrony of activation fails in particular to instantiate even moderately complex relations in the case of variable binding. These examples show that neural models of combinatorial structures can only succeed if they provide a neural instantiation of both the parts and the relations of combinatorial structures.

In computational terms, a blackboard architecture provides a way to instantiate the parts and the relations of combinatorial structures. A blackboard architecture consists of a set of specialized processors (or ‘demons’, Selfridge 1959) that interact with each other by means of a blackboard (or ‘workbench’, or ‘bulletin board’). Each processor can process and modify the information that is stored on the blackboard. In this way, the architecture can process or produce information that exceeds the ability of each individual processor. In the case of language, one could have processors for the recognition of words and (other) processors for the recognition of specific grammatical relations. These processors could then communicate by using a blackboard in the processing of a sentence. Thus, with the sentence *The little star is beside a big star*, the word processors could store the symbol for *star* on the blackboard, the first time in combination with the symbol for *little*, and the second time in combination with the symbol for *big*. Other processors could then determine the relation (*beside*) between these two copies of the symbol for *star*. Jackendoff (2002) discusses blackboard architectures of this kind for phonological, syntactic and semantic structures.

In the next section, I will propose and discuss a neural blackboard architecture for sentence structure based on neural assemblies. To address the problems described by Jackendoff (2002), neural word assemblies are not copied in this architecture. Instead, they are temporarily bound to the neural blackboard, in a manner that distinguishes between different occurrences of the same word, and that preserves the relations between the words in the sentence. For instance, with the sentence *The cat chases the mouse*, the word assembly for *cat* is bound to the blackboard as the subject or agent of *chases*, and the assembly for *mouse* is bound as the object or theme of this verb.

With the neural structure of *The cat chases the mouse*, the architecture can produce correct answers to questions like “Who chases the mouse?” or “Whom does the cat chase?”. Questions like these can be referred to as ‘binding questions’, because they test

the ability of an architecture to ‘bind’ familiar parts in a (potentially novel) combinatorial structure. A neural instantiation of a combinatorial structure such as *The cat chases the mouse* fails if it cannot produce the correct answers to the questions stated above. In language, binding questions in fact query ‘who does what to whom’ information, which is the characteristic form of information provided by a sentence (e.g., Pinker 1994; Calvin & Bickerton 2000). Aphasic patients, for instance, are tested on their language abilities using non-verbal ‘who does what to whom’ questions (e.g., Caplan 1992). In general, the ability to answer binding questions is of fundamental importance for cognition, because it is related with the ability to select information needed for purposive action (e.g., Van der Heijden & van der Velde 1999).

6. A neural blackboard architecture of sentence structure

In line with Pulvermüller (1999), words are assumed to be encoded in terms of neural ‘word’ assemblies, as illustrated in figure 1 (section 2.1.). It is clear that the relations between the words in a sentence cannot be encoded in terms of direct associations between word assemblies. For instance, the association of *mouse-chases-cat* does not distinguish between the sentences *The mouse chases the cat* and *The cat chases the mouse*.

However, relations between words can be encoded, and the problems discussed by Jackendoff (2002) can be solved, if word assemblies are embedded in a neural architecture in which structural relations can be formed between the word assemblies. A neural architecture of this kind can be formed by means of ‘structure’ assemblies that interact with the word assemblies. The structure assemblies provide the possibility to encode different instantiations of the same word assembly (thereby solving the ‘problem of 2’), and they can be used to bind word assemblies in terms of the syntactic structure of the sentence.

Figure 2 illustrates the neural structure of the sentence *The mouse chases the cat* in this architecture. It consists of word assemblies, structure assemblies for noun phrases (NPs) and verb phrases (VPs), gating circuits used for dynamic control, and memory circuits used to bind assemblies temporarily. In figure 2, the assemblies for *mouse* and *cat* are bound to NP assemblies (N_1 for *cat* and N_2 for *mouse*), and the assembly for *chases* is bound a VP assembly (V_1). The structure assemblies are then bound to each other, in a manner that encodes the verb-argument structure of the sentence. For this purpose, each structure assembly in the architecture is composed of a main assembly (N_i for NP assemblies and V_i for VP assemblies) and one or more subassemblies. In figure 2, the NP and VP assemblies have subassemblies for the arguments agent (a) and theme (t)⁶. To encode *cat* as the agent of *chases*, N_1 is bound with V_1 by means of their agent subassemblies. In turn, N_2 and V_1 are bound with their theme subassemblies, to provide the neural structure for *mouse* as the theme of *chases*.

Main assemblies and subassemblies are assumed to have the ability for reverberating activity, in line with the reverberating activity found in the prefrontal cortex (e.g., Fuster 1973; Amit 1995; Durstewitz et al. 2000). As a result, they will remain active for a while after they have been activated, unless they are inhibited. Subassemblies are connected to main assemblies by means of gating circuits, which control the flow of activation within structure assemblies. For instance, a main assembly can be active, but its subassemblies not, or vice versa. The ability to control the internal dynamics of structure assemblies is

of crucial importance for the neural architecture of sentence structure proposed here. Before illustrating this in more detail, I will first discuss the gating and memory circuits used in this architecture.

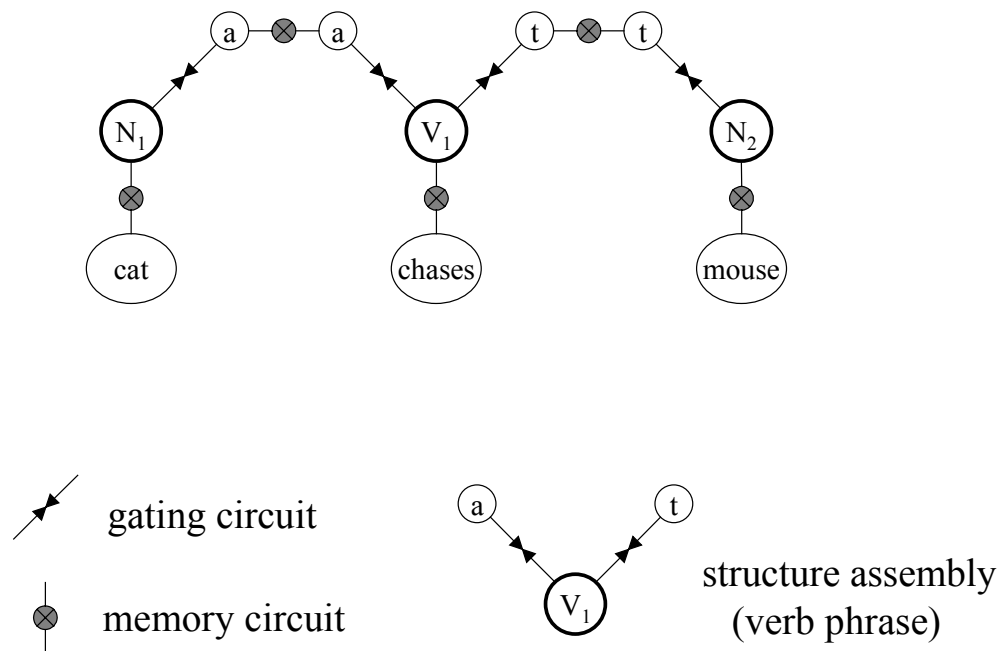


Figure 2. Illustration of the neural sentence structure of *cat chases mouse* in the neural blackboard architecture presented here. The words are encoded with the word assemblies illustrated in figure 1 (section 2.1.). Sentence structure is encoded with ‘structure assemblies’ for noun-phrases (NP assemblies) and verb-phrases (VP assemblies). A structure assembly consists of a main assembly and a number of subassemblies, connected to the main assembly by means of gating circuits. The labeled subassemblies represent the thematic roles of agent (a), and theme (t). Binding between assemblies is achieved with active memory circuits. Here, the assembly for *cat* is bound to the NP assembly N_1 , the assembly for *chases* is bound to the VP assembly V_1 , and the assembly for *mouse* is bound to the NP assembly N_2 . N_1 and V_1 are bound by means of their agent subassemblies and V_1 and N_2 are bound by means of their theme subassemblies.

6.1. Gating and memory circuits

A gating circuit in the architecture consists of a disinhibition circuit, as described by Gonchar and Burkhalter (1999). Figure 3 (left) illustrates a gating circuit in the direction from assembly X to assembly Y . The circuit controls the flow of activation between the two assemblies by means of an external control signal. It operates in the following manner. If the assembly X is active, it activates an inhibition neuron (or group of neurons) i_x , which inhibits the flow of activation from X to X_{out} . When i_x is inhibited by

another inhibition neuron (I_x), that is activated by an external control signal, X activates X_{out} . In turn, X_{out} activates Y . A gating circuit from Y to X operates in a similar manner. Control of activation can be direction specific. Thus, by producing a control signal in the direction from X to Y , activation will flow in this direction (if X is active), but not in the direction from Y to X . The symbol illustrated in figure 3 (left) will be used to represent the combination of gating circuits in both directions (as in figure 2).

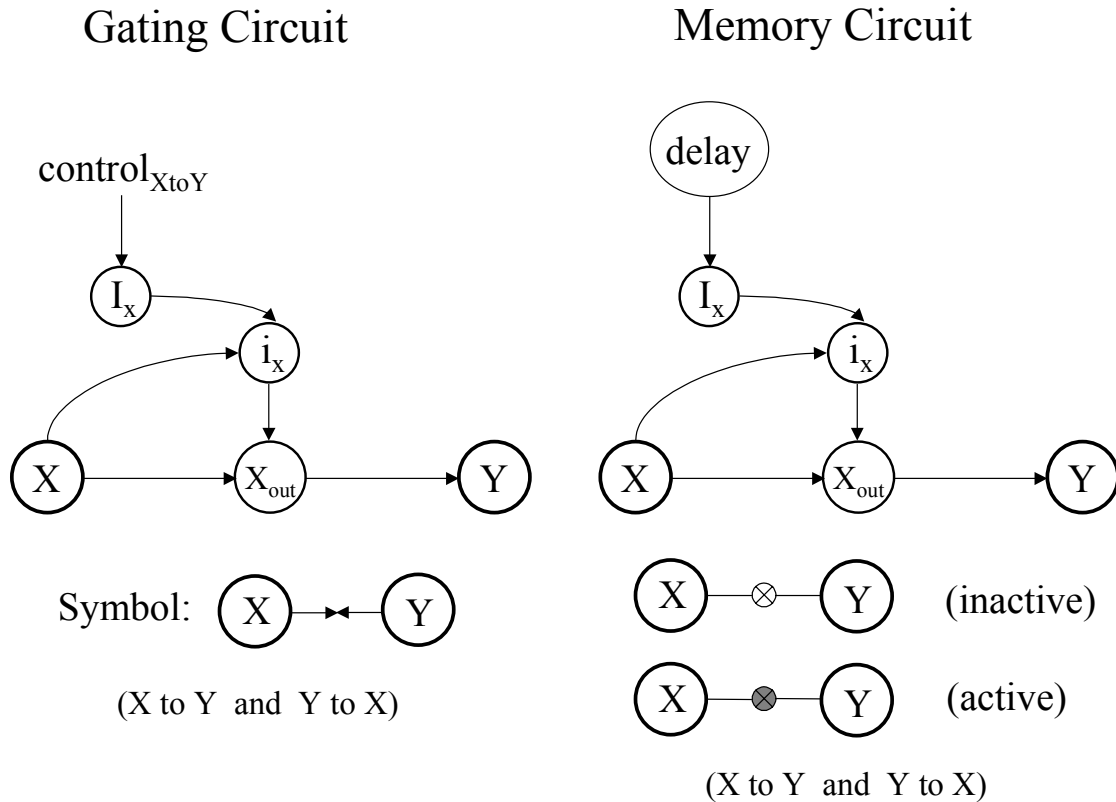


Figure 3. Left: A gating circuit in the direction from assembly X to assembly Y , based on a disinhibition circuit. The large circles depict neural assemblies. The small circles depict (groups of) inhibitory neurons (i). A combination of two gating circuits in the directions X to Y and Y to X is depicted in other figures with the symbol illustrated at the bottom. Right: A memory (gating) circuit in the direction from assembly X to assembly Y , based on a gating circuit with a delay assembly for control. A combination of two memory circuits in the directions X to Y and Y to X is depicted in other figures with the symbols illustrated at the bottom, one for the inactive state and one for the active state of this combined memory circuit.

A memory circuit in the architecture consists of a gating circuit in which the control signal results from a ‘delay assembly’. Figure 3 (right) illustrates a memory circuit in the direction of X to Y . However, each memory circuit in the architecture in fact consists of two such circuits in both directions (X to Y and Y to X). The delay assembly (that controls the flow of activation in both directions) is activated when X and Y are active simultaneously (see below), and it remains active for a while due to the reverberating

nature of the activation in this assembly. As a result, a memory circuit can be in two states: active and inactive. Each state will be represented with the symbol illustrated in figure 3 (right). If the memory circuit is inactive, activation cannot flow between the assemblies connected by the memory circuit. On the other hand, if the memory circuit is active, activation will flow between the assemblies it connects, if one of these assemblies is activated. In this way, an active memory circuit binds the two assemblies it connects. The memory circuits in figure 2 are active, so that word assemblies and structure assemblies are bound in line with the structure of the sentence.

6.2. Overview of the architecture

Figure 4 illustrates the part of the architecture in which nouns can be bound as arguments to verbs. This part is illustrative of the architecture as a whole.

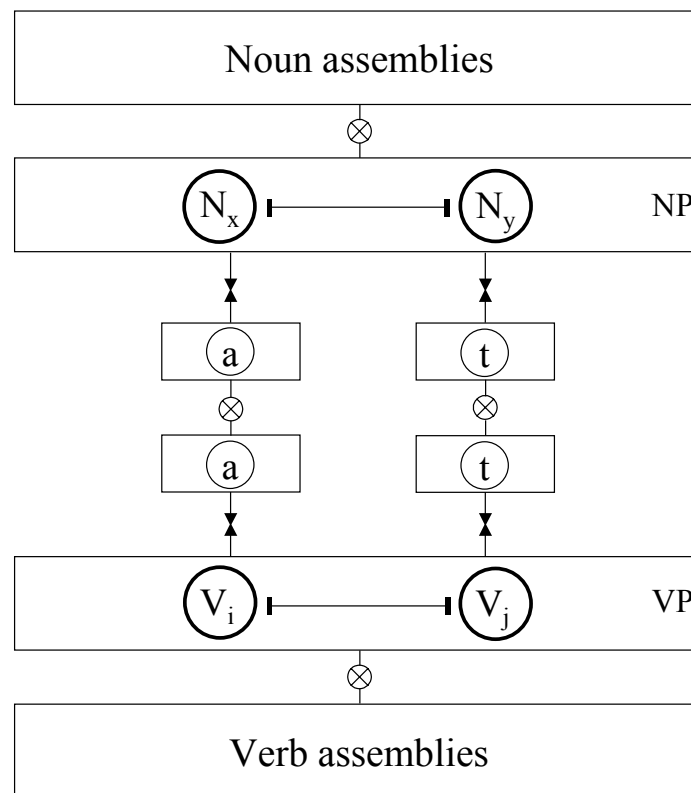


Figure 4. A neural blackboard architecture for verb-argument binding. Word assemblies for verbs are connected to the main assemblies of VP structure assemblies by means of (initially) inactive memory circuits. Word assemblies for nouns are connected to the main assemblies of NP structure assemblies by means of (initially) inactive memory circuits. The agent (a) and theme (t) subassemblies of the VP and NP structure assemblies are connected by means of (initially) inactive memory circuits. Only subassemblies of the same kind are connected to each other. The main assemblies of VP assemblies are mutually inhibitory. Likewise for NP structure assemblies.

Each noun (word) assembly is connected to the main assembly of each NP assembly by means of a memory circuit, which is initially inactive. Likewise, each verb (word) assembly is connected to the main assembly of each VP assembly by means of an initially inactive memory circuit. Main assemblies of the same kind are mutually inhibitory. Each NP and VP main assembly is connected to a number of subassemblies by means of gating circuits. The gating circuits can be activated in a selective manner by neural control circuits (not shown in the figure). For instance, the gating circuits between the main assemblies and the agent subassemblies can be activated without activating the gating circuits for the theme subassemblies. Finally, all subassemblies of the same kind are connected by means of memory circuits. For instance, each agent subassembly of the NP assemblies is connected to each agent subassembly of the VP assemblies by means of an (initially inactive) memory circuit.

In the processing of a sentence it is assumed that one of the NP assemblies will be activated whenever the assembly for a noun is activated. It is arbitrary which of the NP assemblies is activated, provided that the assembly is ‘free’. A structure assembly is ‘free’ when it is not already bound to a sentence structure, that is, when all memory circuits connected with that assembly are inactive⁷. As illustrated in figure 4, only one NP main assembly can be active at the same time, due to the competition between the NP main assemblies that results from their mutual inhibition. It is assumed that the active NP assembly will remain active until a new NP assembly is activated by the occurrence of a new noun in the sentence⁸. The selection of a VP assembly proceeds in the same manner.

In all, in the order of 10^2 VP assemblies and 10^2 NP assemblies would probably be needed in this architecture. When a sequence of structure assemblies has been activated, the first assemblies in the sequence will return to the inactive state (i.e., will again be ‘free’), due to the decay of delay activity in the memory circuits connected with these assemblies. In this way, only a subset of the structure assemblies will be concurrently active in the blackboard architecture.

6.2.1. Connection structure for binding in the architecture

Figure 5 (right) illustrates that the connection structure between the agent subassemblies in figure 4 basically consists of a matrix-like array of ‘columns’. This connection structure is illustrative of every connection between assemblies by means of memory circuits. Each column contains a (combined) memory circuit (figure 3, right), including the delay assembly that can activate the memory circuit. Each column also contains a circuit that can activate the delay assembly (figure 5, left). This circuit is also a disinhibition circuit, in which the delay assembly will be activated if the neurons N_{in} and V_{in} are active at the same time. The neurons N_{in} and V_{in} , in turn, are activated by the respective agent subassemblies of a NP assembly and a VP assembly.

The activated agent subassembly of a given NP assembly activates the N_{in} neurons in a horizontal row of columns (as illustrated with N_x in figure 5, right). Likewise, the activated agent subassembly of a given VP assembly activates the V_{in} neurons in a vertical row of columns (as illustrated with V_i in figure 5, right). The delay assembly in the column on the intersection of both rows will be activated if the agent subassemblies of N_x and V_i are active at the same time. This results in the binding of these agent subassemblies (illustrated with the shaded memory circuit symbol in figure 5, right).

The columns within each horizontal and vertical row (figure 5, right) are mutually inhibitory. Inhibition is initiated by the active delay assemblies⁹ (figure 5, left). Thus, when the agent subassemblies of N_x and V_i are bound by an active memory circuit, the active delay assembly in their mutual column initiates the inhibition of all columns in the same horizontal and vertical row. This prevents a second binding of N_x with another VP assembly, or of V_i with another NP assembly, by means of agent subassemblies.

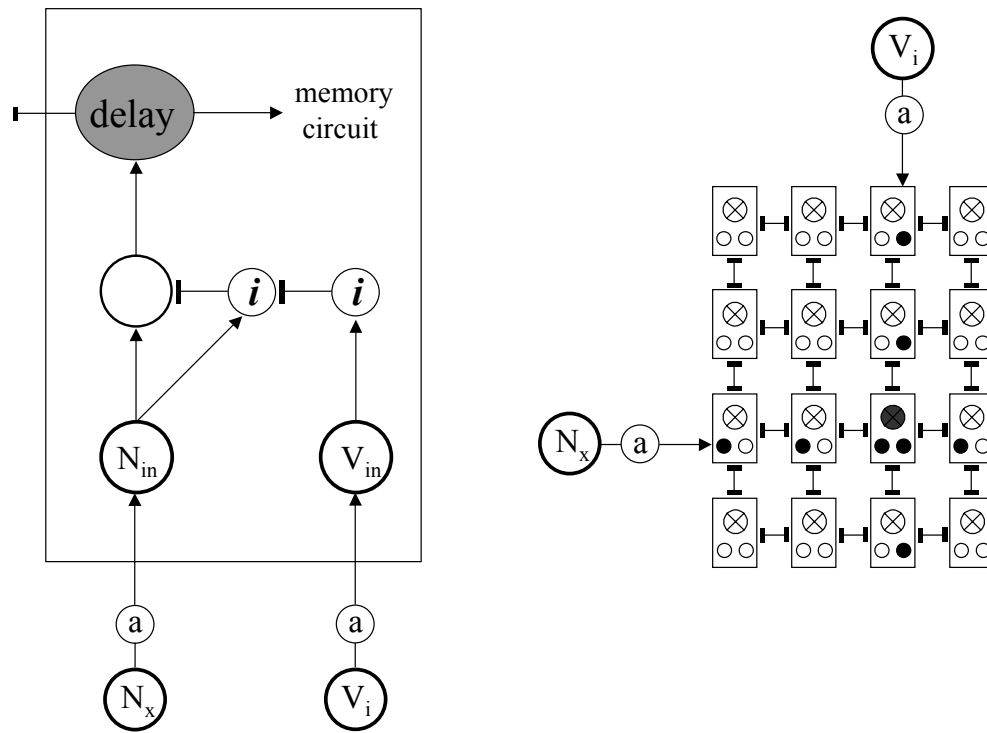


Figure 5. Connection structure for the agent subassemblies in figure 4. Left: a delay assembly in a memory circuit (figure 3, right) is activated when the subassemblies connected by the memory circuit are concurrently active (using a disinhibition circuit). Right: Each agent subassembly of all NP assemblies is connected to each agent subassembly of all VP assemblies with a specific ‘column’ in an array of columns. Each column consists of the memory circuits that connect both subassemblies, together with the circuit in figure 5 (left). The active subassembly of N_x will activate all N_{in} neurons in its horizontal row of columns. Likewise, the active subassembly of V_i will activate all V_{in} neurons in its vertical row of columns. This results in the activation of the delay assembly in the (combined) memory circuit in their corresponding column. Columns in horizontal and vertical rows are mutually inhibitory. Inhibition is initiated by active delay assemblies in the memory circuits.

The binding process of the sentence in figure 2 proceeds as follows. When the assembly for *cat* is activated, a NP assembly is activated as well, and the assembly for *cat* is bound to this NP assembly by the activated memory circuit that connects the two assemblies. In the same manner, the assembly for *chases* will be bound to a VP assembly.

To achieve the binding of *cat* as the agent of *chases*, the gating circuits between the NP and VP main assemblies and their agent subassemblies have to be activated. The active NP and VP main assemblies (i.e., the NP and VP main assemblies bound to *cat* and *chases*) will then activate their agent subassemblies. In line with the process illustrated in figure 5, this will result in the binding of these two agent subassemblies.

In this architecture, it is assumed that gating circuits are activated by neural control circuits. Basically, the control circuits instantiate syntactic operations, based on the active word assemblies and the activation state of the blackboard. In the example above, these circuits will detect that in a sequence like *cat chases* (or *N-V*), *cat* is the agent of the verb *chases*. In response, they will activate the gating circuits for the agent subassemblies of all NPs and VPs. This results in a binding between the active NP assembly and the active VP assembly by means of their agent subassemblies, in the manner discussed above. The binding of *mouse* as the theme of *chases* proceeds in a similar manner. I will discuss the operations of these control circuits in somewhat more detail later on. First, however, I will discuss how this neural blackboard architecture for sentence structure can solve the ‘four challenges for cognitive neuroscience’ presented by Jackendoff (2002), and discussed in section 2.

6.3. Multiple instantiation and binding in the architecture

Figure 6 illustrates the neural structures of the sentences *The cat chases the mouse*, *The mouse chases the cat* and *The cat bites the dog* in the neural blackboard architecture. The words *cat*, *mouse* and *chases* occur in more than one sentence, which creates the problem of multiple instantiation (the problem of 2) for the assemblies of these words. Figure 6 shows that the problem of multiple instantiation is solved by binding each word assembly to a unique structure assembly. For instance, the word assembly for *cat* is bound to the NP assemblies N_1 , N_4 and N_5 . Similarly, different VP assemblies (V_1 and V_2) encode the verb *chases* in different sentences. The different structure assemblies can be bound in line with the structure of each sentence. In this way, *cat* can be the agent of *chases* in one sentence (by binding N_1 and V_1 with their agent subassemblies) and the theme of *chases* in another sentence (by binding N_4 and V_2 with their theme subassemblies). Furthermore, *cat* can also be the agent of another verb (*bites*) in a third sentence, using N_5 . This illustrates how the binding problem in language can be solved (on the level of verb-argument binding), even with multiple instantiations of words.

The internal structure of the NP and VP assemblies, given by the gating circuits, is of crucial importance in this respect. Without this internal structure, the neural structures in figure 6 would collapse into direct associations between neural assemblies, which would result in a failure to distinguish between, for instance, *The cat chases the mouse* and *The mouse chases the cat*. With the control of activation provided by gating circuits, the neural structures of these two sentences can be selectively (re)activated.

6.3.1. Answering binding questions

Selective reactivation of a sentence structure is necessary to retrieve information from the blackboard architecture. In particular, to answer specific binding questions, such as the question “Whom does the cat chase?”. The question provides the information that *cat* is the agent of *chases* and it asks for the theme of *chases* in that sentence (i.e., it asks for x in the sentence *cat chases x*). The production of the answer consists of the selective

activation of the word assembly for *mouse*. This can be achieved through a competition process between the VP assemblies, in which V_1 emerges as the winner. After that, the activation of the gating circuits for theme will result in the activation of N_2 by V_1 , and thus in the activation of *mouse* as the answer to the question¹⁰.

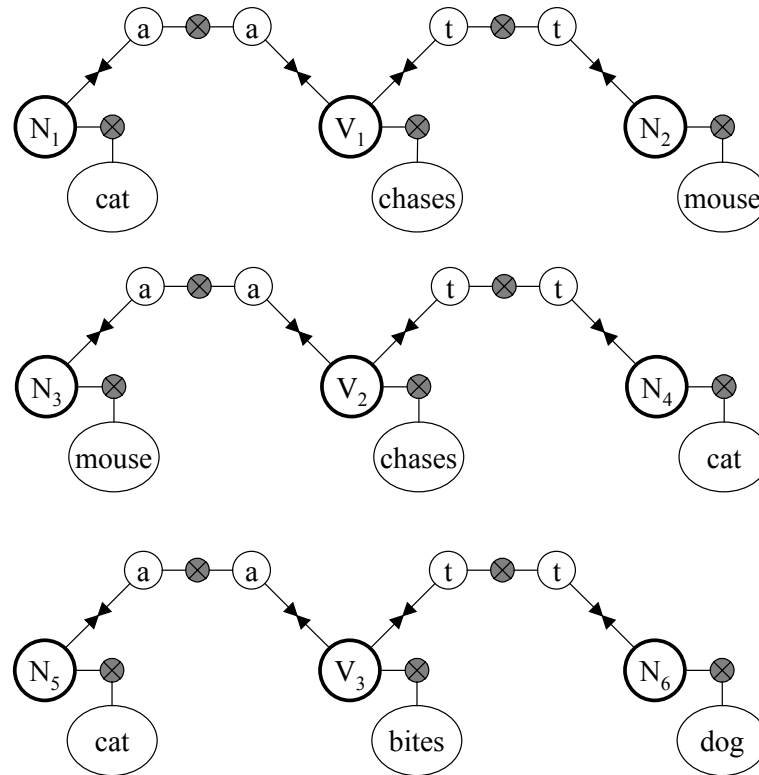


Figure 6. Combined instantiation of the sentences *cat chases mouse*, *mouse chases cat* and *cat bites dog* in the architecture illustrated in figure 4 (ignoring *the* for the moment). The multiple instantiations of *cat*, *chases*, and *mouse* in different sentences (and in different thematic roles) are distinguished by the different NP or VP structure assemblies to which they are bound.

The competition between the VP assemblies is determined by the information provided by the question, which results in the activation of the assemblies for *cat* and *chases*, and the gating circuits for agent. Inspection of figure 6 shows that V_1 receives the most activation in this case, because it is activated by *chases* and N_1 (through *cat*), whereas V_2 is only activated by *chases* and V_3 is only activated by N_5 (through *cat*). The mutual inhibition between VP main assemblies (figure 4) will then result in V_1 as the winner of the VP competition. This analysis has been confirmed by pilot simulations, using populations of spiking neurons to simulate the assemblies (Van der Velde & de Kamps 2003b).

In contrast with the question “Whom does the cat chase?”, the question “Who chases the cat?” will result in a VP competition in which V_2 is the winner. The difference is given by the selective activation of the gating circuits. Both questions result in the

activation of the assemblies for *cat* and *chases*, but they activate different gating circuits. The first question defines *cat* as the agent of *chases*. This information can be used to produce the activation of the gating circuits for agent, which results in V_1 as the winner of the VP competition, as analyzed above. The second question defines *cat* as the theme of *chases*. In this case, the theme gating circuits will be activated, so that N_4 (activated by *cat*) can activate V_2 . This route of activation was blocked in case of the first question. Thus, with the second question, V_2 emerges as the winner because it receives the most activation in this case. Then, *mouse* can be produced as the answer, because the question asks for the agent of *chases* (i.e., it asks for x in x *chases* *cat*).

Figure 6 illustrates that the neural blackboard architecture can solve the binding problem in language on the level of verb-argument binding. However, extensions of the neural blackboard architecture presented thus far are needed to handle the more massive form of binding found in linguistic structures (Jackendoff 2002).

6.4. Extending the blackboard architecture

A first extension of the architecture is introduced in figure 7. As Bickerton argued (Calvin & Bickerton 2000), an important step in the evolution of language consisted of the transformation illustrated in the top-half of figure 7. The diagram on the left is a sentence structure as it occurs in protolanguage (a precursor of modern language), whereas the diagram on the right is a basic sentence structure in modern language (language with syntax, as Bickerton puts it). The difference resides in the fact that one of the arguments of the verb is placed outside the verb's direct influence (i.e., the verb-phrase), and occupies a controlling position of its own in the sentence (as the subject).

The bottom-half of figure 7 shows a similar transition in terms of the neural architecture proposed here. The structure on the left is the sentence structure of figure 2. For convenience, I have introduced a shorthand presentation of this structure in figure 7. The gating and memory circuits are not shown, connected subassemblies are presented as one, and words are simply written close to their structure assemblies. However, the full structure of figure 2 is still implied. The shorthand version does not result in ambiguities: subassemblies are always connected to their main assemblies with gating circuits, subassemblies are only bound to other subassemblies of the same kind (and always with active memory circuits), and word assemblies are always bound to structure assemblies of the corresponding type (e.g., nouns to NP assemblies).

The sentence structure on the left in the bottom-half of figure 7 resembles the diagram on the left in the top-half of the figure. In turn, the sentence structure on the right in the bottom-half of the figure (also in shorthand presentation) resembles the diagram on the right in the top-half of the figure. In this sentence structure, the NP of *cat* is not directly bound to the VP of *chases*. Instead, it is bound to a new sentence structure assembly (S). Binding is achieved through the noun subassembly (n) of the NP assembly (not shown in figure 4), and the corresponding noun subassembly of the S assembly. Likewise, the VP assembly is bound to S with verb subassemblies (v).

The connection structures of the noun subassemblies and the verb subassemblies are similar to the connection structure of the agent subassemblies illustrated in figure 5. Furthermore, the S main assemblies are mutually inhibitory, similar to the NP and VP main assemblies (figure 4). This does not mean that only one sentence at a time could be stored in the blackboard. As illustrated in figure 6, information is stored in the blackboard

by means of active delay assemblies, which are not mutually inhibitory (except in the manner illustrated in figure 5). The inhibitory interaction between main assemblies of the same kind is important for the dynamic control of the activation process in the blackboard when information is stored or retrieved, as in the process of answering binding questions.

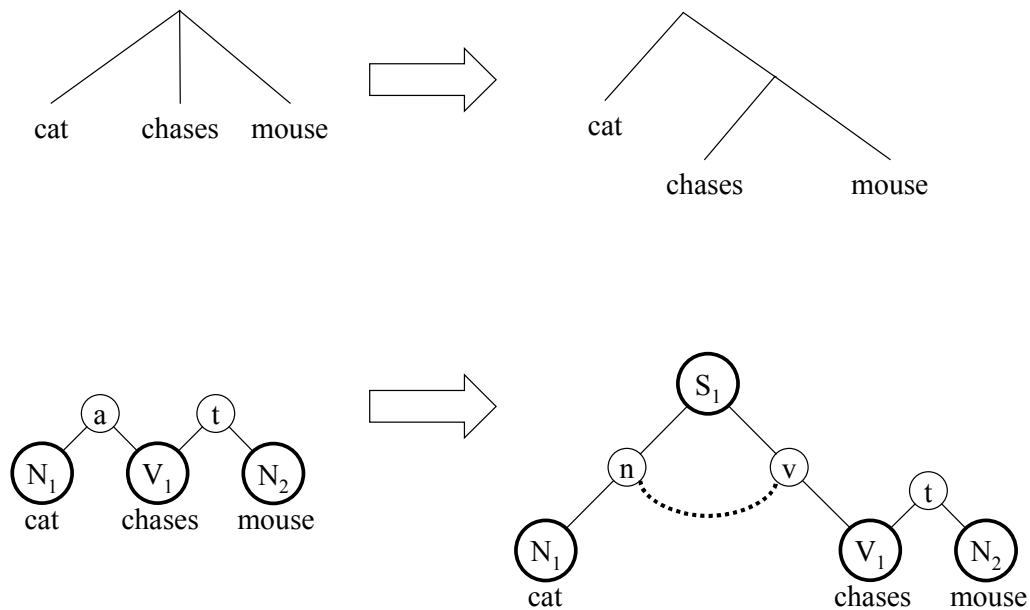


Figure 7. Top: Transformation of sentence structure in proto-language (left) to sentence structure in modern language (right), after Calvin & Bickerton (2000). Bottom: similar transformation in terms of neural sentence structures. The neural sentence structure of *cat chases mouse* on the left is the same as in figure 2, but in a ‘shorthand’ presentation. The neural sentence structure of *cat chases mouse* on the right (also in ‘shorthand’ presentation) consists of a new structure assembly for sentence (S), with subassemblies for noun (n) and verb (v). The dotted line between the noun and verb subassemblies represents the possibility of encoding agreement between subject and verb by means of these subassemblies.

The dotted line between the noun and verb subassemblies indicates that these subassemblies can be used to encode agreement between the subject *cat* and the verb *chases* (as in *cat chases* versus *cats chase*). For instance, S assemblies could have different noun and verb subassemblies for single and plural, which can be activated selectively. Once a noun is bound to the noun subassembly for single, this subassembly will enforce a binding of the verb to a verb subassembly for single as well.

Further extensions of the architecture proceed along similar lines. Basically, they consist of the introduction of new structure assemblies, and new subassemblies needed

for appropriate binding. The architecture can be extended with new structure assemblies because of its modular connection structure. New structure assemblies and the subassemblies needed for binding can be added on without affecting the other connection structures in the architecture.

6.4.1. The modular nature of the blackboard architecture

The modular nature of the blackboard architecture is illustrated in figure 8, which presents the structure for Jackendoff's sentence *The little star is beside a big star*. The new structure assemblies in this sentence structure are determiner assemblies (D_1 and D_2), adjective phrase assemblies (Adj_1 and Adj_2), and prepositional phrase assemblies (P_1). The D_i assemblies are bound to NP assemblies with determiner subassemblies (d), the Adj_i assemblies are bound to NP assemblies with adjective subassemblies (adj), and the P_i assemblies are bound to VP assemblies with preposition-verb subassemblies (pv) and to NP assemblies with preposition-noun subassemblies (pn). The connection structure of each of these new kinds of subassemblies is again similar to the connection structure illustrated in figure 5. Furthermore, main assemblies of the same kind are again mutually inhibitory.

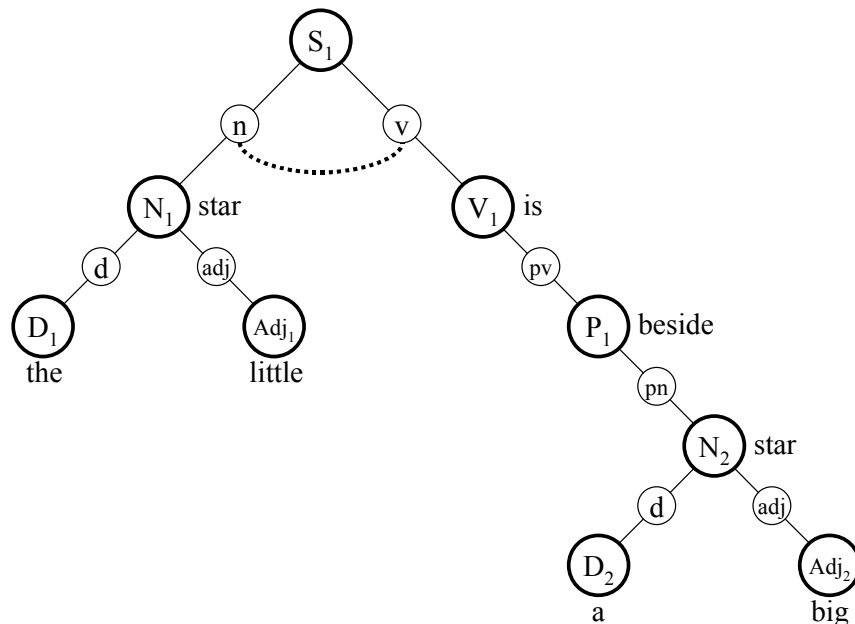


Figure 8. Neural sentence structure for *The little star is beside a big star*. The structure assemblies are similar to those in figure 7 (bottom-right), with new structure assemblies for determiner (D_i), adjective phrase (Adj_i) and prepositional phrase (P_i), and new subassemblies for determiner (d), adjective (adj), preposition-verb (pv) and preposition-noun (pn).

The sentence structure in figure 8 (again) illustrates the solution of the problem of 2 provided by the architecture, and it illustrates the solution of the massiveness of the binding problem in linguistic structures. Even though there is only one word assembly for *star*, it can participate in two different constituents of the sentence, due to the fact that it is bound to two different NP assemblies. Bound to N_1 , *star* has the determiner *the* and the adjective *little*, and it constitutes the subject of the sentence, which is bound directly to S. Bound to N_2 , *star* has the determiner *a* and the adjective *big*, and it is bound to the preposition *beside*, which in turn is bound to the verb of the sentence.

Questions can be again be answered by selectively activating structure assemblies and gating circuits. Consider, for instance, the question “Which star is beside a big star?”. In global terms, the answer can be produced if S_1 is activated (to inhibit other S_i assemblies), so that the gating circuits for the noun subassemblies can be activated (because the question asks for the adjective of the subject). This will result in the activation of N_1 , and, in turn, of D_1 and Adj_1 , which produces the answer *the little star*. S_1 will be activated due to the information *is beside a big star* provided by the question. The phrase *a big star* results in the activation of N_2 , which thus initially wins the competition over N_1 . However, after the selection of S_1 , N_1 will be activated due to the activation of the ‘subject’ gating circuits. Conversely, the question “Where is the little star?” produces the activation of S_1 and V_1 , and it asks for the prepositional phrase of the sentence. The answer will result from activating the gating circuits for the preposition-verb subassemblies.

The introduction of the new structure assemblies in figure 8 raises the question of how many different kinds of structure assemblies would be needed in the neural blackboard architecture. A preliminary answer at this stage is that the architecture would have a particular kind of structure assembly for each kind of constituent that can occur in a linguistic structure. Later on, I will illustrate this point with the encoding of embedded clauses. First, however, the two remaining problems presented by Jackendoff (2002) have to be solved: the problem of variables, and the problem of how a linguistic structure in working memory can be transferred into long-term memory (i.e., constituent binding by means of activation versus constituent binding by means of synaptic modification). I will discuss a solution of these two problems in the next two subsections, beginning with the latter problem.

6.5. Constituent binding in long-term memory

An important role in the storage of information in long-term memory is played by the hippocampus and surrounding areas (or hippocampal complex, Nadel & Moscovitch 2001). The hippocampal complex (HC) has the ability for rapid storage of information by means of synaptic modifications (Rolls & Treves 1998). A prominent view of the role of the HC is given by the ‘Hebb-Marr’ model (McNaughton & Nadel 1990). In this view, HC neurons form a conjunctive encoding of those neurons that are concurrently active in the cortex (Rolls & Treves 1998; O’Reilly & Rudy 2001). The encoding results from the modification of the synapses between the active neurons in the cortex and the active neurons in the HC. These neurons then combine into an auto-associator (Marr) or a cell assembly (Hebb), that can be reactivated as a whole when only a part of it has been reactivated. In this way, the HC forms a ‘snapshot-like’ memory of an event with the duration of about a second (Rolls & Treves 1998). Given the connection structure within

the HC (sparse connectivity), different events or episodes can be separated in memory, because they can be encoded with different and non-overlapping groups of neurons in the HC (O'Reilly & Rudy 2001).

A crucial aspect of encoding with HC neurons is the fact that the storage of information is unstructured (Roll & Treves 1998). That is, the HC acts as a simple binding device, which forms a conjunctive encoding of the input that is concurrently available. However, the HC does not, by itself, encode systematic relations within the input (O'Reilly & Rudy 2001). Therefore, in the words of O'Reilly & Rudy (2001, p. 320): “all relationship information must be present in the inputs to the hippocampus, which can then bind together the relational information with other information about the related items in a conjunction”.

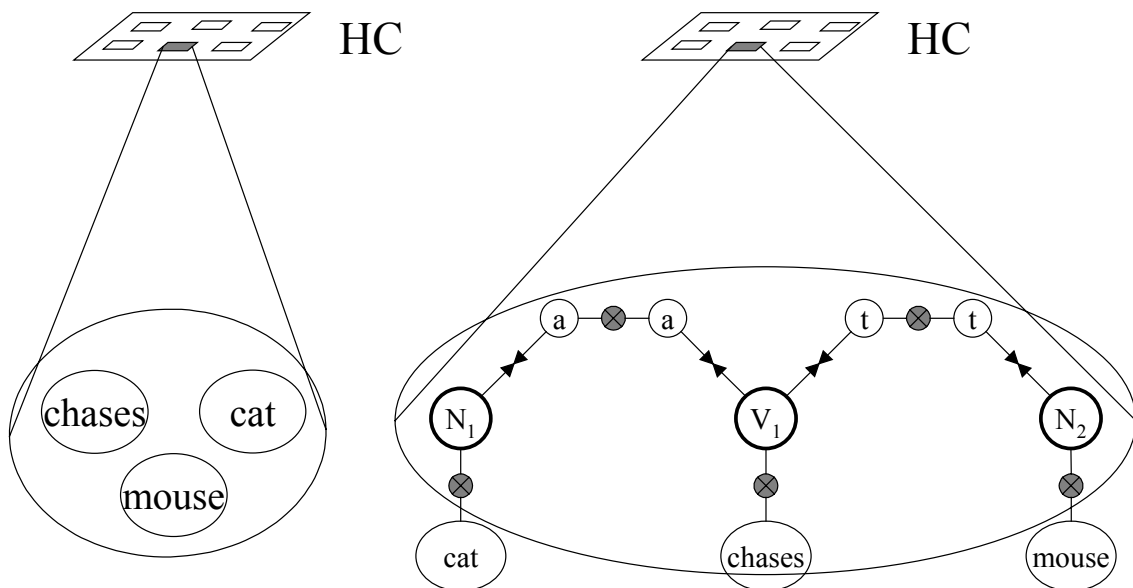


Figure 9. Left: conjunctive encoding of the assemblies for *cat*, *chases* and *mouse* with a neuron (or group of neurons) in the hippocampus complex (HC). Right: conjunctive encoding of the neural sentence structure of *cat chases mouse* with a neuron (or group of neurons) in the hippocampus complex (HC).

Figure 9 (left) illustrates what this means in terms of the word assemblies activated (within a second or so) by the sentence *The cat chases the mouse*. The HC will form a conjunctive encoding of the word assemblies, but not of their relations¹¹. As a result, the same conjunctive encoding of the word assemblies will be formed with the sentence *The mouse chases the cat*. Thus, HC conjunctive encoding of word assemblies creates the familiar binding problem. Reactivation by the HC will reactivate the word assemblies for

cat, *mouse*, and *chases*, but not the structure of either *The cat chases the mouse* or *The mouse chases the cat*.

The problem can be solved by including relationship information in the input to the HC, as described by O'Reilly & Rudy (2001). This will occur if the activity in the neural blackboard architecture for sentence structure is included in the input to the HC, as illustrated in figure 9 (right). In this way, the HC can reactivate a neural sentence structure by reactivating the neural blackboard. Thus, figure 9 (right) illustrates that a neural blackboard architecture plays a crucial role in the storage of combinatorial structures in long-term memory (i.e., in terms of synaptic modification). Even a conjunctive encoding as provided by the HC is sufficient, if the activity in the blackboard is included in the encoding.

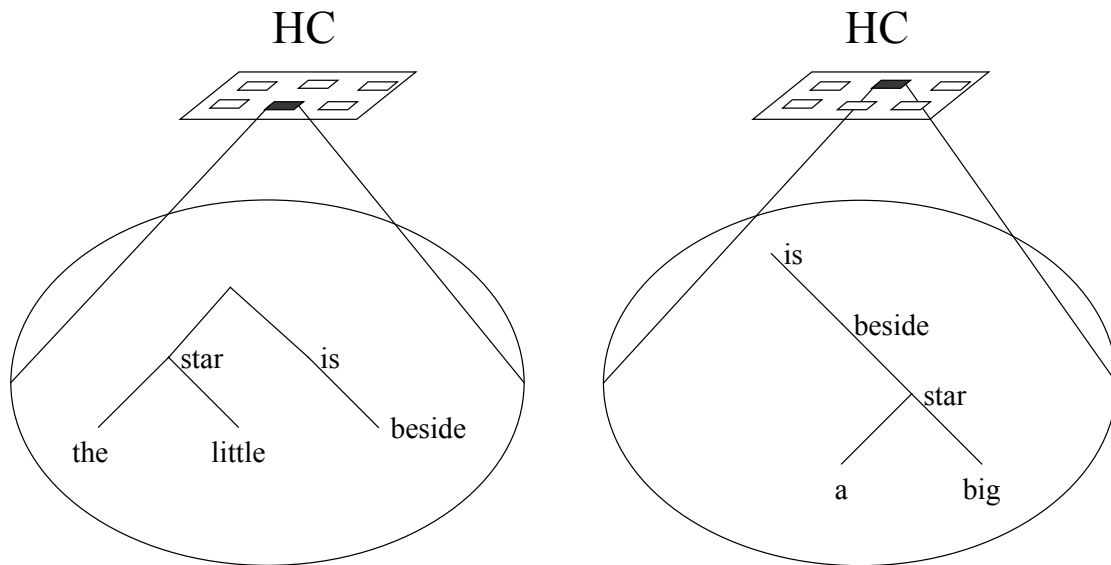


Figure 10. Encoding of the neural sentence structure of *The little star is beside a big star* (figure 8) with two different neurons (or groups of neurons) in the hippocampus complex (HC). Each neuron encodes a part ('episode') of the sentence structure. Both parts can be overlapping.

With longer sentences, it is likely that the HC will encode the sentence structure in terms of a sequence of events, each consisting of a conjunctive encoding of a part of the sentence structure. Figure 10 illustrates this process for the structure of the sentence *The little star is beside a big star* presented in figure 8. Two different HC neurons encode two different parts of the sentence, which could be partly overlapping. The whole sentence structure can be reactivated in the blackboard if one of the HC neurons reactivates the

part of the sentence structure it encodes. The overlap between the two structural parts can then result in the activation of the other HC neuron, which will reactivate the remaining part of the sentence structure.

6.5.1. One-trial learning

In particular, the activity of the delay assemblies in the memory circuits has to be included in the input to the HC, because the structure of a sentence is completely determined by the set of active delay assemblies. In fact, as hinted at in figure 9 (right), the HC encoding would not have to include all (or even any) of the word assemblies of the sentence. The overall structure can be retrieved (i.e., binding questions can be answered) as long as the delay assemblies can be reactivated by the HC (in line with the sentences presented in figure 6).

The fact that HC encoding of the active delay assemblies is sufficient to store the sentence structure in memory constitutes an important aspect of the use of delay activity as a binding mechanism. The delay assemblies in the blackboard can remain active concurrently without causing interference, unlike the word and structure assemblies. The reverberating activity of delay assemblies will then provide sufficient time for the process of synaptic modification to proceed. This solves the problem of one-trial learning, as described by Jackendoff (2002, p. 66): “It is usually argued that transient connections have the effect of gradually adjusting synaptic weights (so-called Hebbian learning). But what about cases in which one trial is sufficient for learning? For example, you say to me, *I’ll meet you for lunch at noon*. I reply, *OK*, and indeed I do show up as agreed. My long-term memory has been laid in on the basis of one trial; there hasn’t been any opportunity to adjust synaptic weights gradually”.

Figure 9 (right) illustrates how one-trial learning can proceed by means of the blackboard architecture. The word assemblies in *The cat chases the mouse* are indeed activated briefly, to prevent the interference effects that would otherwise occur. But the delay assemblies can remain active for a longer period, because they do not interfere with each other. This provides the opportunity to adjust the synaptic weights between the HC and the delay assemblies gradually, in line with Hebbian learning. In this way, a long-term memory of a sentence structure can be formed on the basis of one trial.

6.5.2. Explicit encoding of sentence structure with synaptic modification

Although the conjunctive encoding of the blackboard by the HC provides an encoding of sentence structure in terms of synaptic weights, retrieval of information from long-term memory still requires that the blackboard activation of the sentence structure is reactivated by the neurons in HC, probably in a sequence as illustrated in figure 10. However, one could imagine that a more explicit encoding of sentence structure in terms of synaptic weights would be possible, which on its own could be used to retrieve information. An important function of the HC is indeed to provide a quick but temporal storage of information, so that the interaction between the HC and the cortex can result in a (slower) transference of that information to the cortex, where it can be incorporated in the existing knowledge base (e.g., O’Reilly & Rudy 2001). After such a process, a sentence structure could be encoded explicitly in the cortex in terms of synaptic modification.

Figure 11 presents a neural structure of *The cat chases the mouse* in terms of synaptic modification (the structure in the brackets represents the shorthand version). As in figure 2, the structure consists of word assemblies, structure assemblies, and the appropriate bindings between the assemblies. The word assemblies in figure 11 are the same as those in figure 2. The structure assemblies in figure 11 are of the same kind as those in figure 2 (NP and VP). Structure assemblies in figure 11 also consist of main assemblies and subassemblies, connected with gating circuits. However, binding in figure 11 is not achieved by memory circuits (as in figure 2), but instead consists of synaptic modification. In this way, the word assemblies are directly connected to the main assemblies. Subassemblies of the same kind are also directly connected to each other, effectively forming a single assembly.

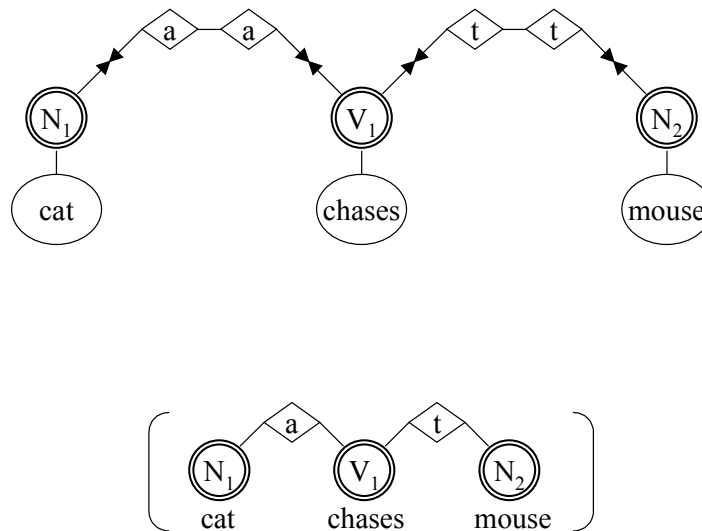


Figure 11. Explicit encoding of neural sentence structure in long-term memory, illustrated with the sentence *cat chases mouse*. Word assemblies are bound to main assemblies of structure assemblies with synaptic modification, with nouns to noun phrase (NP) assemblies and verbs to verb phrase (VP) assemblies. Subassemblies of the same kind are bound with synaptic modification. This effectively results in a single subassembly, as illustrated with the agent (a) and theme (t) subassemblies of NP and VP assemblies. A ‘shorthand’ presentation of the sentence structure is given in brackets.

The structure assemblies in figure 11 do not belong to the blackboard architecture to which the structure assemblies in figure 2 belong (i.e., the architecture of figure 4). Binding in the architecture of figure 4 is always temporary, depending on the activity of

the delay assemblies in the memory circuits. When the delay assemblies in the memory circuits connected to a structure assembly are no longer active, the structure assembly can be used to encode a different sentence structure (again temporarily). This characteristic is the basis of the productivity of the architecture in figure 4. With the ability to reuse the structure assemblies, the architecture can encode arbitrary and novel sentence structures on the fly.

In contrast, the structure assemblies in figure 11 cannot be reused in this way. Due to binding with synaptic modification, the structure in figure 11 is of a more permanent nature, created specifically for this particular sentence structure. As a result, a knowledge base of this kind consists of a collection of sentence structures ('facts'), each one created after a prolonged learning process, in line with the transference of information between the HC and the cortex discussed above. Thus, it is possible that the sentence *The cat chases the mouse* belongs to this knowledge base, but the sentence *The mouse chases the cat* does not.

6.6. Variable binding

The knowledge base illustrated with the sentence structure in figure 11 can be used in a rule-based derivation with variable binding, such as the derivation that *own(Mary, book)* follows from *give(John, Mary, book)*. Here, I will discuss how the binding question "What does Mary own?" can be answered on the basis of the fact (proposition) *John gives Mary a book and Mary gives John a pen*. In section 3, I argued that the model of Shastri and Ajjanagadde (1993), based on synchrony of activation, is faced with serious difficulties in the case of such a proposition, due to the multiplication of the arguments *John* and *Mary* in different roles in the proposition (i.e., the problem of 2).

Figure 12 shows how the combination of the facts *John gives Mary a book* and *Mary gives John a pen* will be encoded in terms of the neural structure introduced in figure 11 (using the shorthand presentation). The verb *give(x,y,z)* has three arguments (agent, recipient, and theme), thus the VP and NP assemblies have an additional subassembly for recipient (r). The word assembly for *give* is connected to two VP main assemblies (V_1 and V_2), which are mutually inhibitory. V_1 is bound to the NP assemblies for *John* (N_1), *Mary* (N_2), and *book* (N_3), in the manner that it encodes the fact *give(John, Mary, book)*. Similarly, V_2 is bound to the NP assemblies for *Mary* (N_4), *John* (N_5), and *pen* (N_6), in the manner that it encodes the fact *give(Mary, John, pen)*.

Even though the fact *Mary owns a book* does not belong to the knowledge base, the question "What does Mary own?" can be answered on the basis of the fact *John gives Mary a book* if the information provided by the question is transformed into information related with *give(x,y,z)*. The question "What does Mary own?" provides the information that *Mary* is the agent of *own*, and it asks for the theme in the proposition. In short, the question provides information of the form *own(Mary, ?)*. In terms of *give(x,y,z)*, the question provides the information that *Mary* is the recipient of *give*, and it asks for the theme in the proposition. In short, the question provides information of the form *give(-, Mary, ?)*. In general, information of the form *own(X,?)* can be transformed into information of the form *give(-,X,?)* on the basis of a long-term association between *own-agent* and *give-recipient* (as in the model of Shastri and Ajjanagadde, 1993).

In line with the process of answering binding questions (section 6.3.1.), the information of the form *own(X, ?)* will produce the activation of the assembly for *own*

and the gating circuits for agent. In contrast, the information of the form $give(-,X, ?)$ will produce the activation of the assembly for $give$ and the gating circuits for recipient. Therefore, the activation produced by $own(X,?)$ cannot be concurrently active with the activation produced by $give(-,X,?)$. In figure 12, this would result in the activation of $give$ and the combined activation of the gating circuits for agent and recipient. The VP assemblies V_1 and V_2 would then receive an equal amount of activation when the assembly for X (*Mary*) is active, so that *book* and *pen* would have an equal probability of being produced as the answer to the question.

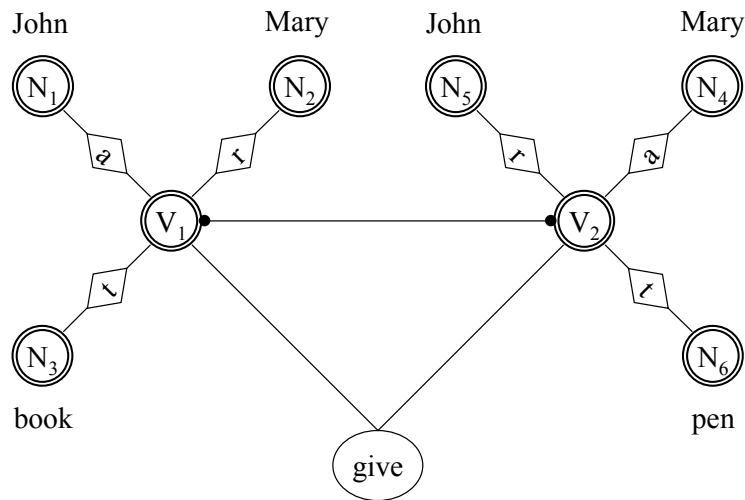


Figure 12. The explicit encoding of the (combined) neural structures of *John gives Mary a book* and *Mary gives John a pen* in long-term memory, in the manner of the structure presented in figure 11 (with ‘shorthand’ presentation). The subassemblies include a new subassembly for recipient (r). VP main assemblies are mutually inhibitory.

Concurrent activation produced by $own(X,?)$ and $give(-,X,?)$ would be prevented if the activation produced by $own(X,?)$ consists of an ‘attractor’ state (Amit, 1995) of a control network, which is associated with the attractor state in the control network produced by $give(-,X,?)$. First, the control network will be in the attractor state related with $own(X,?)$. But when an answer is not produced in this way (because $own(Mary, book)$ does not belong to the knowledge base), the attractor state in the control network would change into the associated attractor state that corresponds with $give(-,X,?)$ ¹².

When the information related with $give(-, Mary, ?)$ is singled out, the answer can be produced by activating $Mary$ and $give$, and the gating circuit for recipient. As illustrated in figure 12, this will result in V_1 as the winner of the competition between the VP assemblies. After, that, the answer can be produced by activating the gating circuits for theme.

The transformation of the information related with $own(X, ?)$ into information related with $give(-, X, ?)$ does not depend on $Mary$, or on any of the other variables in figure 12 (i.e., $book$, pen , or $John$). It only depends on the association between $own-agent$ and $give-recipient$. Thus, the derivation of $own(Mary, book)$ from $give(John, Mary, book)$ is a rule-based derivation with variable binding. The same process can operate on the blackboard architecture in figure 4, so that a novel structure like $give(Dumbledore, Harry, broom)$ can result in the answer to the question “What does Harry own?”.

6.6.1. Neural structure versus spreading of activation

In the neural structure illustrated in figure 12, the fact $give(John, Mary, book)$ can be used to answer the question “What does Mary own”, even though the fact $give(Mary, John, pen)$ is also instantiated in the architecture. The two facts do not interfere, because the gating circuits control the flow of activation in the structure assemblies.

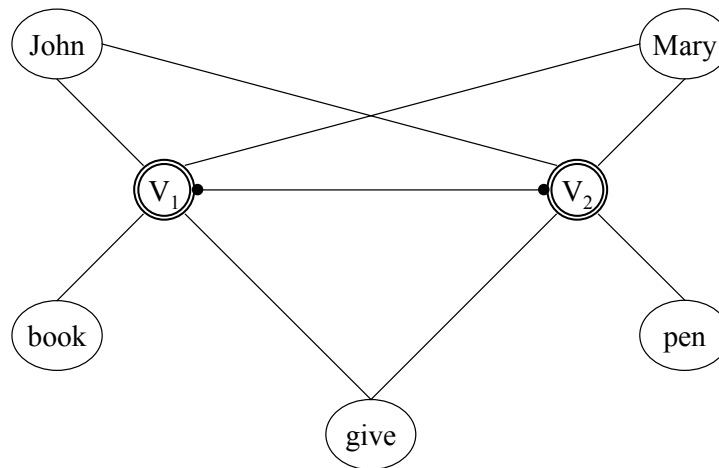


Figure 13. Illustration of the collapse of the neural structures presented in figure 12 when the gating circuits are removed. The result is a network of assemblies based on spreading of activation.

Figure 13 shows the collapse of the structure presented in figure 12 when the gating circuits are removed, to illustrate the importance of activation control provided by the gating circuits in the neural structures presented here. Without the gating circuits, a main assembly and its subassemblies merge into a single assembly. In fact, the NP assemblies can be omitted altogether, because the word assemblies for the nouns are now directly connected with the VP main assemblies V_1 and V_2 . Because all assemblies are now directly connected with excitatory or inhibitory connections, processing only depends on spreading of activation. The information related with *give(-, Mary, ?)* results in the activation of the assemblies for *give* and *Mary*. Due to the uncontrolled spreading of activation, the activation of *give* and *Mary* results in an equal activation of V_1 and V_2 , so that a correct answer to the question cannot be given without ambiguity or error.

In fact, any question will result in ambiguities or error in this uncontrolled spreading of activation network. For instance, a question like “Who gives a book?” will result in the activation of both *John* and *Mary* as potential answers, even though V_1 will win the competition over V_2 . In contrast, in the structure in figure 12, the question “Who gives a book?” will result in *John* as the answer, because the question will result in the activation of the gating circuits for agent after V_1 has won the VP competition.

6.7. Structural dependencies in the blackboard architecture

As Jackendoff (2002) noted, a solution of the ‘four challenges for cognitive neuroscience’ would allow a more productive interaction between neural network modeling and linguistic theory to begin. To illustrate the possibility of such an interaction, I will discuss the neural blackboard structures of the sentences (1) and (2), discussed in section 4.2. They are repeated here for convenience:

- The cat that the dog that the boy likes bites chases the mouse* (1)
The fact that the mouse that the cat chases roars surprises the boy (2)

In section 4.2, I argued that these two sentences pose a problem for models that process sentences in terms of word strings or strings of word category labels (*N-V* strings). Both sentences have the same word category structure (*N-that-N-that-N-V-V-V-N*), but they are different in terms of complexity (Gibson 1998), with (1) rated as far more complex than (2). The difference in complexity between the sentences is related with the different bindings between the constituents in both sentences. As such, they form an interesting example of the massiveness of the binding problem that occurs in language.

A neural instantiation of sentence structure has to account for the differences in constituent binding illustrated with sentences (1) and (2), as any linguistic theory of sentence structure would have to do. But a neural instantiation of sentence structure should also provide an explanation of the observed differences in complexity between these sentences (and other performance effects, Van der Velde 1995).

The structural difference between sentences (1) and (2) is related with the embedded clauses they contain. The instantiation of clauses in the architecture is discussed below.

6.7.1. Embedded clauses in the blackboard architecture

Figure 14a presents the structure of the sentence *The cat that bites the dog chases the mouse* (without the determiners *the*). This sentence contains the subject-relative clause *that bites the dog*. To encode and bind this clause, a new clause structure assembly (C) is introduced, with a new clause subassembly (c).

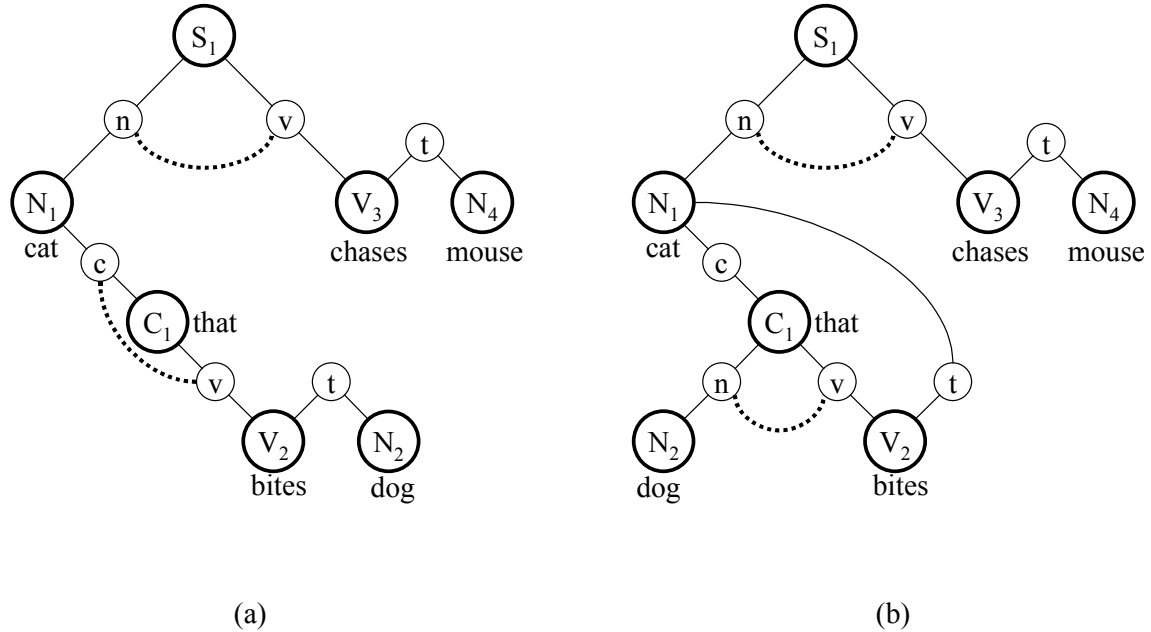


Figure 14. (a). Illustration of the neural sentence structure of *The cat that bites the dog chases the mouse* (without the determiners). The structure is based on the sentence structure presented in figure 7 (bottom-right), with the addition of a clause structure assembly (C) and a clause subassembly (c). The dotted lines represent agreement between subject and verb. (b). Illustration of the sentence structure of *The cat that the dog bites chases the mouse*, using the same kind of structure assemblies as in (a).

C assemblies play a role in the encoding of a clause that is similar to the role played by S assemblies in the encoding of the main sentence (*cat chases mouse* in figure 14a). Nevertheless, they are a different kind of structure assemblies. Unlike S assemblies, C assemblies have to be bound to one of the structure assemblies in the sentence, as illustrated with the binding between C₁ and N₁ in figure 14a. This requirement entails a structural difference between C and S assemblies. In particular, it requires a new kind of subassembly (c), which is used to bind C assemblies to other assemblies. Furthermore, the word assemblies of complementizers can bind with C assemblies, as illustrated with *that* in figure 14a¹³.

Like S assemblies, C assemblies can be used to encode agreement between subject and verb. In the case of a subject-relative clause, agreement exists between subject of the main sentence (*cat*) and the verb of the relative clause (*bites*), as indicated with the dotted line between the clause and verb subassemblies of the C assembly in figure 14a.

Figure 14b presents the structure of the sentence *The cat that the dog bites chases the mouse*. This sentence contains the object-relative clause *that the dog bites*. Encoding and binding is achieved with the same kind of structure assemblies as in figure 14a. However, in this case, *dog* is the subject of *bites*, so it is bound to the noun subassembly of C_1 , and agreement between *dog* and *bites* is encoded by agreement between the noun and verb subassemblies of C_1 , as indicated with the dotted line.

In an object-relative sentence like *The cat that the dog bites chases the mouse*, the subject of the main sentence is the theme of the verb in the relative clause. In the structure presented in figure 14b, this is instantiated by the binding of *cat* (N_1) with *bites* (V_2) by means of their theme subassemblies. This poses a problem for the control of binding in this sentence. When V_2 is active, N_2 is the active NP assembly, not N_1 . Therefore, the theme subassembly of N_1 has to be activated before the activation of N_2 (a subassembly can remain active even if its main assembly is deactivated). Thus, the gating circuits for theme have to be activated before the activation of the main assembly of N_2 .

With the object-relative sentence in figure 14b, the control circuits could conclude from the sequence *cat that dog* (or *N that N*) that *cat* is the theme of the next verb, so that the gating circuits for theme have to be activated before the activation of N_2 . This control of activation is not needed for the subject-relative sentence in figure 14a. Furthermore, the structure in figure 14b requires an additional binding (between C_1 and N_2 with noun subassemblies). These activation differences between the structures in figure 14b and figure 14a could be the basis for the fact that object-relative sentences are more difficult to process than subject-relative sentences (Gibson 1998).

6.7.2. Multiple embedded clauses

Figure 15a presents the structure of *The cat that the dog that the boy likes bites chases the mouse* (1). Sentence (1) contains the double center-embedded object-relative clause *that the dog that the boy likes bites*. Sentences of this type are notoriously hard to process, to the point that they can be classified as unprocessable (Gibson 1998).

The encoding of the phrase *The cat that the dog* proceeds in the same way as in figure 14b, so that the theme subassembly of N_1 (*cat*) will be activated to bind with the theme subassembly of the next verb. However, another embedded clause is introduced, instead of a verb. The phrase *the dog that the boy* is structurally similar to the phrase *the cat that the dog*, so that the theme subassembly of N_2 (*dog*) will be activated to bind with the theme subassembly of the next verb. Thus, when the first verb (*likes*) appears, there are two subassemblies that can bind with the theme subassembly of this verb, whereas the verb should bind with *dog* (N_2) as its theme argument. The situation is similar with the second verb (*bites*), which should bind with *cat* (N_1) as its theme argument. The two problematic bindings are indicated with the dashed lines in figure 15a.

Figure 15b shows the structure of the sentence *The fact that the mouse that the cat chases roars surprises the boy* (2). The structure of (2) is very similar to the structure of (1), except for the fact that *roars* (V_2) does not have a theme argument. A phrase beginning with *The fact that* will be interpreted as a complementary clause, so that the

theme subassembly of N_1 (*fact*) will not be activated. When the object-relative clause in *the mouse that the cat chases* appears, the theme subassembly of N_2 (*mouse*) will be activated to bind *mouse* as the theme of *chases* (as in figure 15a). However, in contrast with the structure of (1) in figure 15a, *mouse* (N_2) can bind with the first verb (*chases*) as its theme because the theme subassembly of N_2 is the only active theme subassembly at that moment.

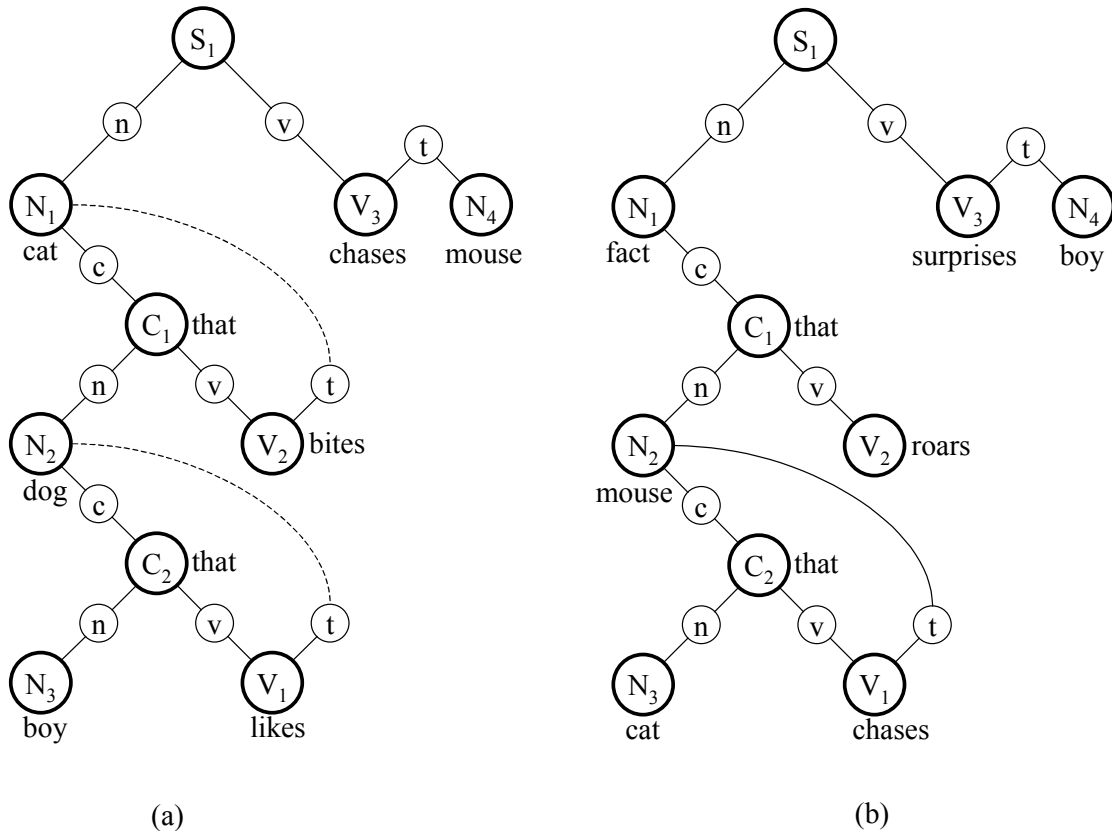


Figure 15. (a). Illustration of the neural sentence structure of *The cat that the dog that the boy likes bites chases the mouse*, with the same kind of structure assemblies as used in figure 14b. (b). Likewise, the neural structure of the sentence *The fact that the mouse that the cat chases roars surprises the boy*.

Thus the difference in complexity between (1) and (2) results from a difference in structural dependency between both sentences. In (1) the subject of the main sentence (*cat*) is also the theme of a verb in an object-relative clause. In combination with the second object-relative clause, this results in an ambiguity of the binding of *cat* (N_1) or *dog* (N_2) as the theme of *likes* (V_1) or *bites* (V_2). In contrast, in (2) the subject of the main clause (*fact*) is not bound to any of the verbs in the embedded clauses, so that the ambiguities in (1) do not arise in (2).

At face value, the binding problem that arises with the theme subassemblies of the sentence structure in figure 15a would also have to arise with the verb subassemblies in both sentence structures in figure 15, in particular for the verb assemblies connected to

the C assemblies (the verb assembly of the S assembly could be activated after the binding of C assemblies has been completed). The activation of C_2 will inhibit the activation of C_1 in both sentence structures, thus the verb subassembly of C_1 has to be activated before C_2 is activated. But the first verb in the sentence (*likes* or *chases*) has to be bound to C_2 , which requires the activation of the verb subassembly of C_2 as well. However, the binding problem with the verb subassemblies can be solved in terms of the dynamics of the binding process, as discussed below.

6.7.3. Dynamics of binding in the blackboard architecture

The binding of subassemblies (of the same kind) occurs in a connection structure as illustrated in figure 5. Figure 16 illustrates the process of subassembly binding between two arbitrary structure assemblies A and B.

In figure 16a, the subassembly of A_{i-1} has activated its horizontal row of columns in the connection structure. If a B_j subassembly would activate its vertical row of columns in the connection structure, a binding would result between A_{i-1} and B_j , in the manner as discussed in section 6.2.1. However, the subassembly of A_i is activated first, which results in the activation of a second horizontal row of columns.

In figure 16b, the subassembly of B_j activates its vertical row of columns in the connection structure. At this moment, a conflict arises between the binding of A_{i-1} with B_j and the binding of A_i with B_j . Due to the inhibitory interaction between the columns in the vertical row of B_j (initiated by activated delay assemblies), only the stronger of these two bindings will survive. Figure 16c illustrates that A_i will bind with B_j if the activation in the horizontal row of A_i is stronger than the activation in the horizontal row of A_{i-1} .

When the binding process of A_i and B_j has been completed, the columns in the horizontal row of A_i (and the vertical row of B_j) will be inhibited due to the active delay assembly in the column that binds A_i with B_j . However, as illustrated in figure 16d, the columns in the horizontal row of A_{i-1} are still active (with the exception of the column in the vertical row of B_j). Thus, the subassembly of A_{i-1} can bind with another B subassembly if that is activated.

The process illustrated in figure 16 shows that two A subassemblies can bind in sequence with B subassemblies if there is a clear difference in activation strength between the two A subassemblies. In that case, the stronger activated A subassembly will bind with the first active B subassembly and the other A subassembly will bind with the second B subassembly. In theory, one could have a whole series of A subassemblies that can bind in sequence with B subassemblies, if the A subassemblies have distinguishable differences in their activation strengths.

Pulvermüller (2000) suggested that a gradual decay of activation in reverberating assemblies (such as the delay assemblies in the memory circuits) could form the basis of a neural pushdown stack. Figure 16 illustrates this possibility. If the subassemblies of A_{i-n} to A_i have been activated that order, and if the activation strength of the subassemblies decays over time, then the subassembly of A_i would have the strongest activation and it would bind to the first B subassembly, as illustrated in figure 16. Then, the subassembly of A_{i-1} would bind to the next B subassembly, as illustrated in figure 16d. In the same manner, all the subassemblies of A_{i-n} to A_i would bind to B subassemblies in the reverse order of their activation, in line with the notion of a pushdown stack¹⁴.

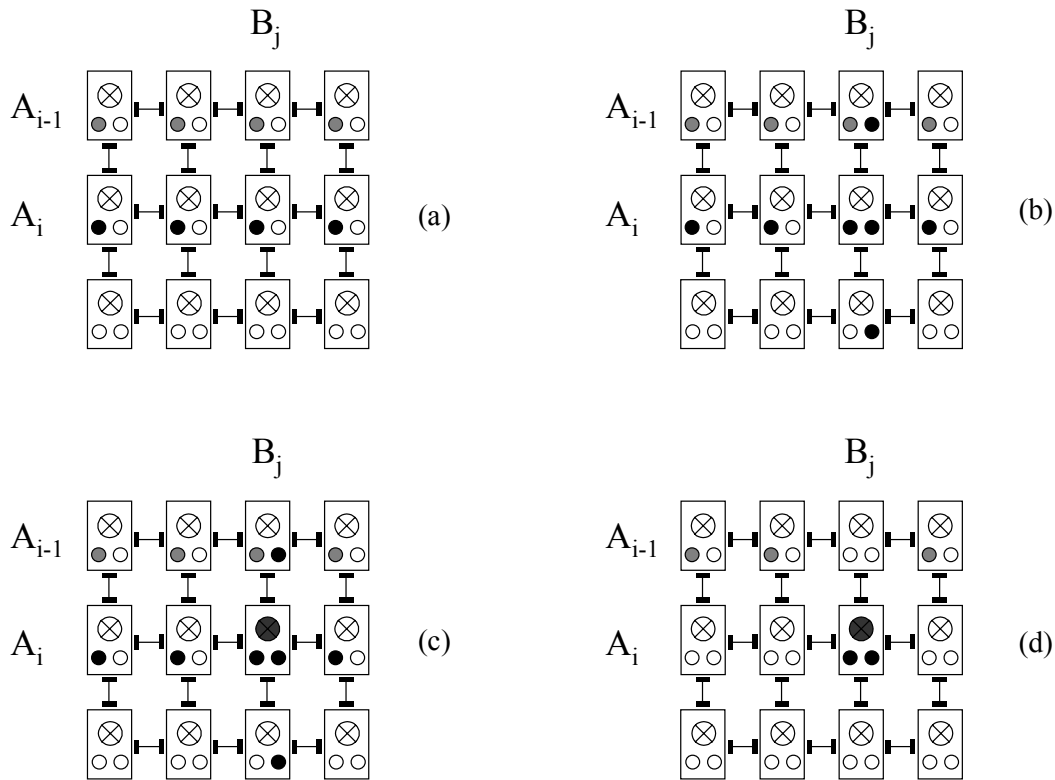


Figure 16. Four stages in the process of subassembly binding between arbitrary structure assemblies A and B, with the connection structure as illustrated in figure 5. (a). The subassemblies of A_{i-1} (first) and A_i (second) have activated their horizontal row of columns. (b). The subassembly of B_j has activated its vertical row of columns. (c). Binding occurs between A_i and B_j , because the activation in the row of A_i is stronger than the activation in the row of A_{i-1} . (d). After completion of the binding process in (c), A_{i-1} can bind to another B assembly. In this way, the connection structure can operate as a pushdown stack.

It is not clear whether such a distinctive and reliable decay of reverberating activity will be found in the brain, due to the fluctuations that can occur in this kind of activity (Amit 1989). However, in one circumstance one can find a clear difference in activation strength between reverberating assemblies. Fuster et al. (1985) investigated the relation between reverberating activity in the prefrontal cortex and the visual cortex. First, they identified neurons in both areas of the cortex that responded to the same objects and that maintained their activation in a delay period. Then, they applied a technique of reversible cooling to one of the areas involved. In this way, the activity of the neurons in that area can be blocked temporarily, but the activity will reappear when the temperature is increased to a normal level. Fuster et al. (1985) observed that blocking the activity of neurons in one area also reduced the activity of the neurons in the other area. The activity in the second area increased again when the activity in the first area reappeared (by terminating the cooling in that area).

6.7.4. Dynamics of binding and complexity

The results of Fuster et al. (1985) indicate that reverberating activity in a neural assembly is stronger when the assembly also receives activation from outside. In this way, the binding of the verb subassemblies in the sentence structures in figure 15 can be explained. The main assembly of C_2 is active when the first verb (*likes* or *chases*) appears. Therefore, the verb subassembly of C_2 is also activated by the main assembly, unlike the verb subassembly of C_1 . As a result, the activity of the verb subassembly of C_2 is stronger than the activity of the verb subassembly of C_1 . In line with the binding process illustrated in figure 16, the verb subassembly of C_2 will bind with the verb subassembly of V_1 (*likes* or *chases*), and the verb subassembly of C_1 will bind with the VP assembly (V_2) of the next verb (*bites* or *roars*).

In contrast, the main assembly of N_2 in figure 15a is not active, due to the activation of N_3 (*boy*), which is needed to bind *boy* with *likes*. Without a clear distinction in activation strength between the theme subassemblies of N_1 and N_2 , the binding process illustrated in figure 16 will not succeed, which results in the complexity associated with sentence (1).

The influence of the dynamics of binding on complexity in the blackboard architecture is further illustrated in figure 17. Figure 17a presents the structure of the sentence:

The cat that the dog that you like bites chases the mouse (3)

Sentence (3) is similar to sentence (1) with the exception of the indexical pronoun (*you*) instead of *boy*. Yet, with an indexical pronoun, sentences of the type (3) are rated as far less complex (Gibson 1998). Indexical pronouns (*I*, *you*) play a distinctive role in language because they introduce the speaker and hearer in a conversation. In this way, they could make a different demand on working memory compared to regular nouns, which explains the reduced complexity of (3) versus (1) in Gibson's (1998) complexity theory.

In terms of the blackboard architecture, indexical pronouns could have their own kind of structure assemblies, in line with the suggestion made in section 6.4.1. With this assumption, sentence (3) will have the neural structure presented in figure 17a. The difficulties with the binding of the theme subassemblies of N_2 (*dog*) and N_1 (*cat*) in the structure presented in figure 15a do not arise here. In the structure presented in figure 17a, N_2 is the last activated NP assembly, so that the main assembly of N_2 is active when the first verb (*likes*) appears. In line with the process illustrated in figure 16, N_2 (*dog*) will bind with V_1 (*like*) as its theme, and N_1 (*cat*) will bind with V_2 (*bites*) as its theme.

Figure 17b presents the structure of the German sentence (4), ignoring the adjective *schlechte* (*bad*):

Der Bauer der die Kuh die schlechte Milch gab schlachtete ist krank (4)

(The farmer who the cow which bad milk gave killed is sick)
(The farmer who killed the cow which gave bad milk is sick)

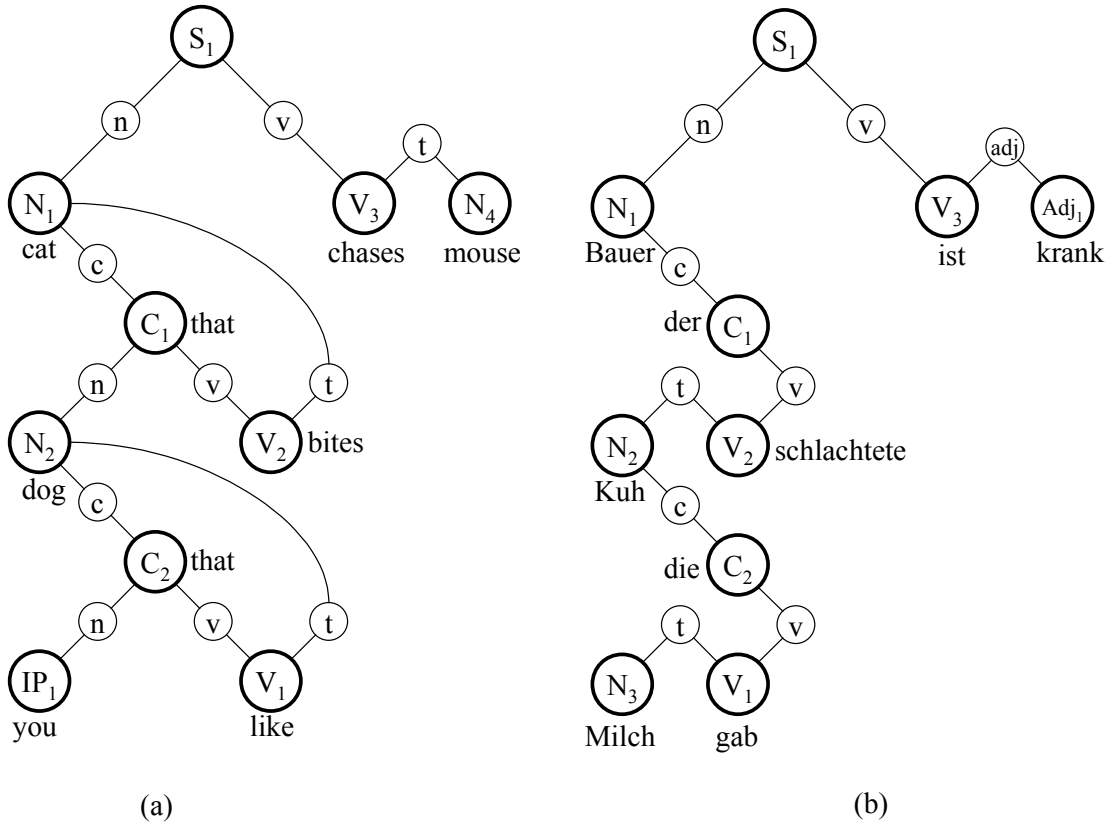


Figure 17. (a). Illustration of the neural sentence structure of *The cat that the dog that you like bites chases the mouse*, with the same kind of structure assemblies as presented in figure 15a, and a new structure assembly for indexical pronoun (IP). (b). Illustration of the neural sentence structure of the German sentence *Der Bauer der die Kuh die (schlechte) Milch gab schlachtete ist krank*, with the same kind of structure assemblies as presented in figure 15a.

The complexity of double center-embedding that occurs in (1) does not arise with this sentence, even though (4) is also a sentence with a double center-embedding (Hawkins 1994; Lewis 1999). However, in contrast with (1), the embedded clauses in (4) are subject-relative clauses. As a result, the problems related with the binding of the theme subassemblies in (1) do not appear in (4). The difference is that in figure 17b the theme subassemblies of N_2 and N_3 have to bind to a verb, instead of the theme subassemblies of N_1 and N_2 in figure 15a. Consequently, the main assembly of N_3 (*Milch*) is active when the first verb V_1 (*gab*) appears, so that N_3 will bind with V_1 as its theme. After that, N_2 (*Kuh*) will bind with V_2 (*schlachtete*) as its theme.

In the sentence structures presented in figure 15 and 17, sentence complexity results from binding problems that arise when a number of structure assemblies of the same kind have to bind in sequence with the overall sentence structure. The nature of these problems is in line with the notion of similarity-based interference as the basis of sentence complexity (Lewis, 1999).

6.8. Further development of the architecture

The neural blackboard architecture for sentence structure outlined here provides a solution to the ‘four challenges for cognitive neuroscience’ presented by Jackendoff (2002). The discussion in section 6.7 also illustrates that the architecture can potentially account for structural and performance aspects of language processing. However, further research is clearly needed to provide a more complete fulfillment of this potential. A few directions of further research can be indicated with the architecture presented thus far.

One line of research would concern the development of the architecture, both in terms of evolution and in terms of growth and learning. In terms of evolution, an important issue is the development of the connection structure presented in figure 5. A benefit of an explicit model as the one in figure 5 is that the model can be used as a target in computer simulations. Thus, starting with more elementary structures, one could investigate whether such a connection structure could develop in an evolution-like process. In terms of growth and learning, an important issue is the question of how specific bindings with connection structures like the one in figure 5 could develop. That is, assuming that an undifferentiated connection structure exists for undifferentiated assemblies, one can investigate whether a learning process could reorganize the undifferentiated connection structure into a connection structure in which distinctions are found between different kinds of structure assemblies and subassemblies. Furthermore, one could investigate whether different languages used in the learning process would result in a different reorganization of the initial connection structure.

Another line of research concerns the instantiation of the neural circuits that control the binding process in the architecture. As noted earlier, the control circuits instantiate basic syntactic operations. Thus, they will be sensitive to the coding principles used in languages to express structural information, like word order, or case marking in languages with free word order (Van Valin 2001). However, the neural control circuits will also be sensitive to the pattern of activation that arises in the blackboard during sentence processing. Figure 16 provides an illustration. An active subassembly produces a significant amount of activation in its connection structure (i.e., its row of columns), which provides the information that a specific binding is required. This information can be used by the control circuits to initiate the activation of a subassembly of the same kind (e.g., a VP theme subassembly when a NP theme subassembly active). In this way, the neural control circuits are engaged in a form of pattern recognition and pattern completion, in which the current state of activation in the blackboard together with the active word assemblies constitute the input pattern and the new state of activation in the blackboard constitutes the output pattern. Pattern recognition is a core capability of networks (Bechtel & Abrahamsen 2002). The fact that a neural blackboard architecture of sentence structure could transform syntactic operations into forms of pattern recognition is an attractive prospect of further research.

A third line of research is the relation between the architecture for sentence structure and other architectures for combinatorial structures in language (Jackendoff 2002) and cognition in general. For instance, words can have an internal structure of their own, which does not seem to agree with word encoding by means of (unstructured) word assemblies (Bierwisch 1999). However, the word assemblies used here can be seen as the interface between word structure and sentence structure. That is, a word assembly is the part of a neural word structure that connects (or ‘anchors’) that structure within the

sentence structure. An example is given in figures 2 and 11, in which the assemblies for *cat*, *chases* and *mouse* form the interface between sentence structures in working memory (figure 2) and sentence structures long-term memory (figure 11). In a similar manner, word assemblies could form the interface between sentence structures and cognitive structures outside language, such as structures in visual cognition. This issue is addressed in more detail below.

7. Neural blackboard architectures of combinatorial structures in vision

The aim of this paper is to show that combinatorial structures can be encoded in neural terms by means of neural ‘blackboard’ architectures. Although combinatorial structures are the ‘quintessential property’ of language (Pinker 1998), they can also be found in visual cognition. Therefore, I will briefly discuss neural blackboard architectures of combinatorial structures in visual cognition, in particular for binding visual object features like shape, color and (relative) location¹⁵. Furthermore, I will discuss how the architectures for visual cognition and language can be combined in a combinatorial structure like *The little star is beside a big star*.

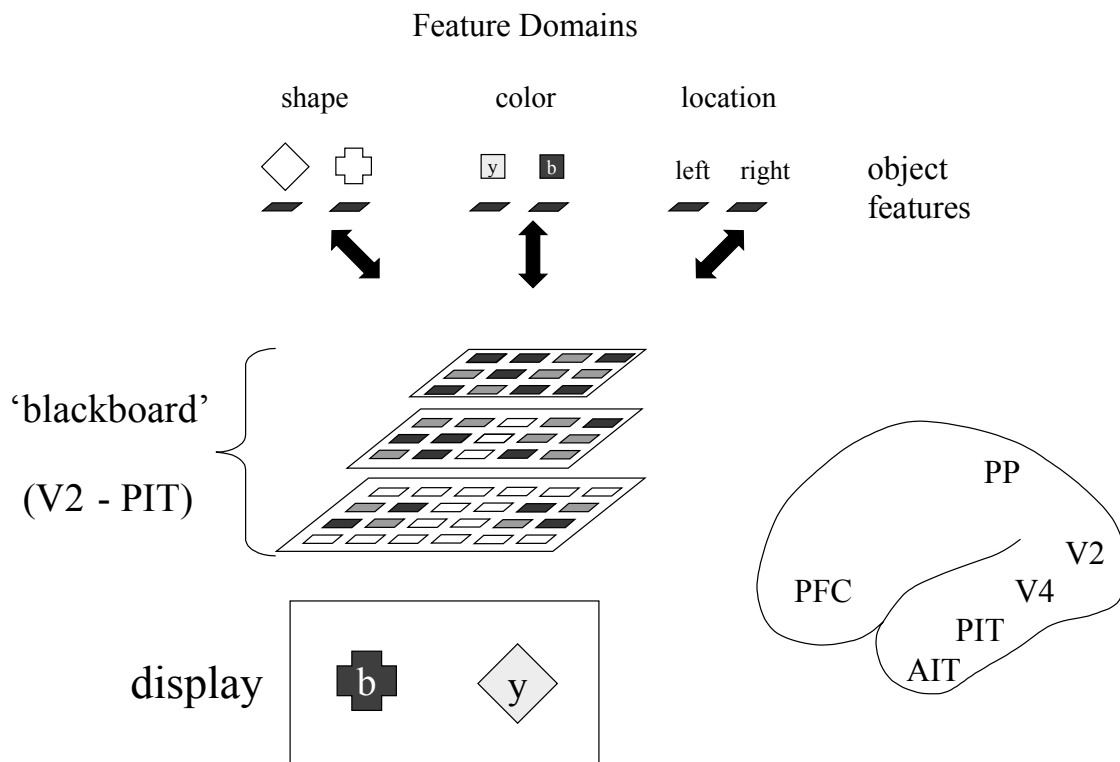


Figure 18. A neural blackboard architecture of combinatorial structure in visual cognition. The ‘blackboard’ consists of the retinotopic areas in the visual cortex (e.g., V2 to PIT). Information about visual features (color, form, motion, location) is processed in feedforward pathways leading to ‘feature domains’ in specialized areas in the visual cortex (e.g., AIT for shape information, PP for location information). In turn, the feature domains send information to the retinotopic areas by means of feedback connections. (AIT = anterior infero-temporal cortex, PIT = posterior infero-temporal cortex, PFC = prefrontal cortex, PP = posterior parietal cortex).

In a blackboard architecture for visual cognition, one would have processors for the recognition of shape, color, location and other visual object features. Combined, these processors would correctly process a visual display of objects, such as a blue cross on the left and a yellow diamond on the right, if they could communicate with each other through a blackboard. In this way, the architecture could answer binding questions like "What is the color of the cross?" or "What is the shape of the yellow object?".

A neural blackboard architecture for combining visual object features in this manner is illustrated in figure 18. The architecture is based on the pathways that determine the structure of the visual cortex (e.g., Livingstone & Hubel 1988; Felleman & van Essen 1991; Oram, & Perrett 1994; Farah et al. 1999). The ventral pathway in the visual cortex includes the areas V2, V4, the posterior inferotemporal cortex (PIT) and the anterior inferotemporal cortex (AIT). This pathway is involved in the processing and selection of 'object features' (e.g., shape and color). Objects are identified through a feedforward network of areas, going from the primary visual cortex (V1) to the higher areas in the temporal cortex (e.g., AIT). The network gradually transforms retinotopic encoding in the lower areas (e.g., V2 to PIT) into a location-invariant identity (e.g., shape, color) encoding in the higher areas (e.g., AIT). The dorsal pathway in the visual cortex leads to the posterior parietal cortex (PP). This pathway is involved in the processing and selection of spatial information (e.g., location of objects) and spatial transformations (e.g., for making eye movements). Both pathways start from the primary visual cortex (V1), but they are also interconnected on the levels of V2, V4 and PIT. Both pathways project to the prefrontal cortex.

Figure 19 (left) illustrates how the shape and the color of two objects, a blue cross and a yellow diamond, would be processed in this architecture. After the primary visual cortex V1 (not shown), the features are processed initially in a feedforward manner (Oram & Perrett 1994). Each object produces a pattern of distributed activation in the areas V2 to PIT that corresponds to the retinotopic location of the object. The activated neurons could respond to one feature (e.g., shape) or to conjunctions of features, like conjunctions of elementary shapes and color (Motter 1994).

The retinotopic object information in the lower layers is gradually transformed into location invariant information, due to the increase in the receptive field size from layer to layer (illustrated with the cones in figure 19). Furthermore, feature encoding is separated in the higher levels of the architecture, where distinctions are made between, for instance, color encoding (e.g., blue vs. yellow) and shape encoding (e.g., cross vs. diamond). The distinctions between object features at this level form the basis for the constituents (parts) that are used to identify combinatorial visual structures.

In human cognition, object features as illustrated in figures 18 and 19 form the basis for conceptual knowledge (e.g., Barsalou 1999; Barsalou et al. 2003). Human language provides ample evidence for the ability to encode object features like shape and color separately, that is, independent of any conjunction of these features. For instance, we can use a word (e.g., *red*) to instruct a viewer to select an object in a visual display based on its color, irrespective of its shape or location (e.g., see Van der Heijden et al. 1996).

7.1. Feature binding

Figure 19 illustrates the binding of shape and color in the blackboard architecture (binding of other features proceeds in a similar manner). The shape of the cross is given as a cue, for instance by the binding question "What is the color of the cross?". The

binding process in the architecture consists of an interaction between a feedforward network and a feedback network.

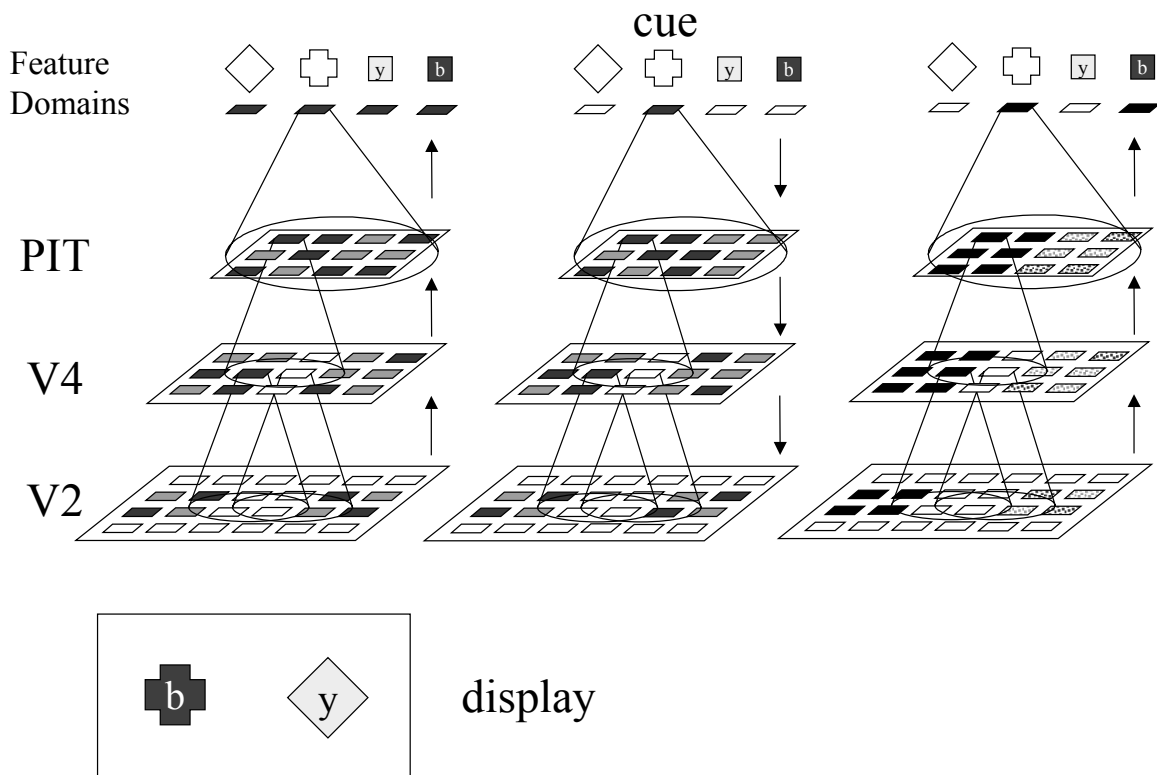


Figure 19. The process of answering the binding question “What is the color of the cross?” in the neural blackboard architecture of figure 18. Left: The shapes, cross and diamond, and the colors, blue (b) and yellow (y), of two objects are processed in feedforward pathways in the retinotopic areas. The receptive field size of neurons increases in higher areas (as indicated with the cones), until encoding is location invariant in the feature domains. Middle: The shape of the target object (the cross) is selected as a cue in the shape feature domain. The selected cue initiates feedback activation in the retinotopic areas. Right: Interaction between feedforward and feedback activation in the retinotopic areas results in the selection (enhancement) of the activation related with the target object in these areas. In turn, this results in the selection of the other features of the target object (its color in this example) in the feature domains. In this way, the features of the target object (‘cross’ and ‘blue’) are bound by the interaction in the neural blackboard architecture.

The feedforward network (figure 19, left) processes the visual display, which results in the identification of the features of the cross and the diamond in the feature domains. The activation pattern in the feedforward network that produces object identification is object selective. That is, when an object is presented on a particular location in the display, it produces a pattern of (distributed) activation in the retinotopic areas in the feedforward network. This pattern of activation is sufficiently different from the pattern of

activation produced by another object, presented on the same location in the display. (Otherwise, a selective identification of the object could not succeed.)

The feedback network in figure 19 (middle) carries information about the selected feature (cue) from the feature domains back to the lower retinotopic areas in the architecture. The feedback network should be seen as lying ‘on top’ of the feedforward network. That is, neurons in the retinotopic areas of the feedforward network have corresponding neurons in the retinotopic areas of the feedback network. The corresponding neurons in both networks could belong to different layers of the same cortical column. Feedback connections are found between almost all areas in the visual cortex (e.g., Felleman & van Essen 1991).

Through the feedback connections, information processed at the level of object features (figure 18) can interact with information processed in the lower retinotopic areas. Thus, the blackboard nature of the visual cortex, as discussed here, basically results from the feedback connections in the visual cortex (Van der Velde 1997; Bulier 2001). The activation patterns in the feedback network are also object selective. This can be achieved by adapting the connections in the feedback network with Hebbian learning, using the selective activation patterns in the feedforward network that occur in the process of object identification (Van der Velde & de Kamps 2001).

The cue-related information in the feedback network (figure 19, middle) interacts with the processing of the display in the feedforward network (figure 19, left). The interaction enhances (‘selects’) the neural activation related with the cue (cross) in the retinotopic areas. The selection of cue-related activation results from the match between the object selective activation in the feedforward network and the object selective activation in the feedback network, as produced by the Hebbian learning procedure in the feedback network described above. The enhanced (selected) cue-related activation in the retinotopic areas can be used to select the color (and the other object features) of the cued object (cross) in the feature domains, as illustrated in figure 19 (right).

The process illustrated in figure 19 shows that the basis for the blackboard architecture in figure 18 is given by the interaction between the retinotopic areas, in which elementary information about the features of an object is combined, and the feature domains, in which identity information of object features is separated. In this way, the visual features of an object can be bound in a combinatorial manner by selecting one of the features (e.g., its shape or color) in the feature domains. Using the interaction process described above, the activation related with the selected object feature will be enhanced in the retinotopic areas. In turn, this enhanced activation will produce the selection of the other features of the object in the feature domains. In particular, a novel combination of familiar visual features (e.g., a purple cow¹⁶) can be identified in this way. A similar process could occur in visual working memory as well.

7.2. A neural blackboard architecture of visual working memory

Neuroimaging studies in humans have shown overlapping areas of activation in the prefrontal cortex (PFC) with spatial and object memory tasks (e.g., Prabhakaran et al. 2000; D’Esposito 2001). Neurons that selectively respond to both identity and location information have been found in monkey PFC as well (Rao et al. 1997; Rainer et al 1998). These results indicate an integrative role of (lateral) PFC in memory tasks (Fuster 2001; Duncan 2001).

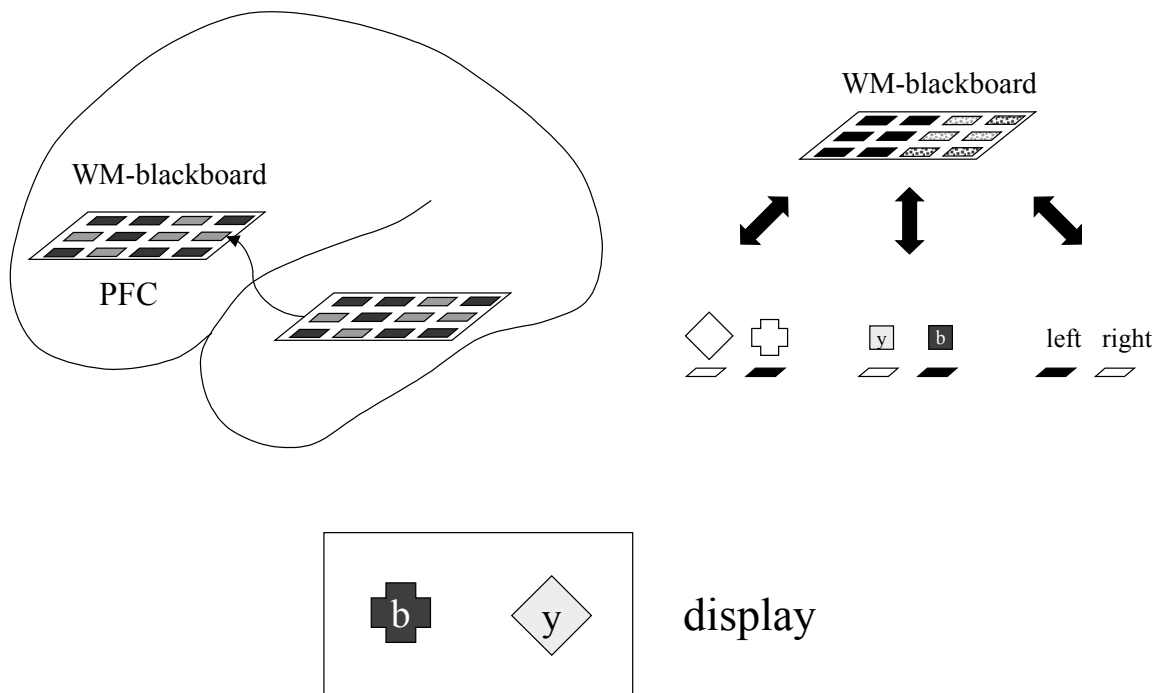


Figure 20. Left: A putative relation between the neural blackboard architecture in the visual cortex and a neural blackboard architecture in visual working memory (WM) in the prefrontal cortex (PFC). Right: An interaction between feature domains and the WM blackboard in PFC can be used to bind the features of an object (e.g., ‘cross’, ‘blue’, and ‘left’) in working memory, similar to the binding process illustrated in figure 19.

A combined selectivity to spatial and object information in PFC is in line with the notion of a blackboard architecture for visual working memory. The neurons in a blackboard visual working memory will respond selectively to combined (elementary) object and location information, similar to the neurons in the retinotopic areas of the visual cortex. Figure 20 (left) illustrates a putative connection between both blackboard architectures. One or more areas in the blackboard of the visual cortex (e.g. PIT) could be connected with a ‘working memory’ (WM) blackboard in lateral PFC. A display of objects could then be encoded in both areas in a similar manner. The difference between the two areas will be found in the nature of the activation. Whereas the activation in the blackboard of the visual cortex results from the processing of the visual display, the activation in the WM blackboard is a form of self-sustained or reverberating activity, in line with WM activity found in PFC (Fuster 1995; Amit 1995; Durstewitz et al. 2000).

7.2.1. Feature binding in visual working memory

The reverberating activity in the WM blackboard can be used to retrieve (select) and bind the features of the objects in a visual working memory task. Figure 20 (right) illustrates that selection and binding of features (again) results from interactions between a blackboard and neurons that encode object features. These neurons could be located in PFC as well (e.g., Wilson et al. 1993), but they could also consist of the neurons that encode object features in the visual cortex. In the latter case, visual working memory will consist of interactions between neurons in PFC and neurons in posterior visual areas (Ruchlin et al. in press).

The nature of the WM blackboard produces the behavioral effects reported by Luck & Vogel (1997). They observed that the number of objects that can be maintained in working memory is limited, but the number of their features is not. In terms of the WM blackboard, too many objects in a display will cause an interference between their distributed activations in the WM blackboard. This interference results in a limitation of the number of objects that can be maintained in working memory (Van der Voort van der Kleij et al. 2003). However, the number of features for each object is not limited. That is, all features of an object can be selected by means of the interaction with the blackboard (figure 20, right) as long as the object activations in the WM blackboard do not interfere.

7.3. Feature binding in long-term memory

Feature binding in visual working memory, as described above, is instantiated in terms of the sustained activation in the WM blackboard. As discussed in section 2.4, this raises the question of how feature binding can be achieved in terms of synaptic modification, which forms the basis of long-term memory. In the case of linguistic structures, this question was answered in terms of the process illustrated in figure 9. The answer proceeds along similar lines for visual feature binding.

Figure 21 (left) illustrates the role of the HC in the case of visual features. A neuron in the HC forms a conjunctive encoding of the object features that are activated by a display of two objects (a blue cross on the left and a yellow diamond on the right). In this way, the neurons that encode the object features can be reactivated when the neuron in the HC is reactivated. However, it is clear that the conjunctive encoding by the HC neuron results in the familiar binding problem (Von der Malsburg 1987), because the relations between the object features are lost in this form of encoding. A display of, say, a yellow cross and a blue diamond (on any of the two locations) would activate the same object features, and would thus be encoded in the same way by the HC neuron as the display in figure 21.

However, as in figure 9, the relations between the object features can be encoded by the HC neurons, if a neural blackboard (e.g., the WM blackboard) is included in the conjunctive encoding, as illustrated in figure 21 (right). In this case, the relationship information is part of the input to the HC (as described by O'Reilly & Rudy 2001), so that the HC can encode the relationship information (the blackboard) together with the object features. When the HC neurons reactivate the blackboard and the object features, the relations between the features of the objects in the display can be retrieved in the manner illustrated in figure 20. The encoding of different events (episodes) in this architecture can proceed in a manner similar to the process illustrated in figure 10.

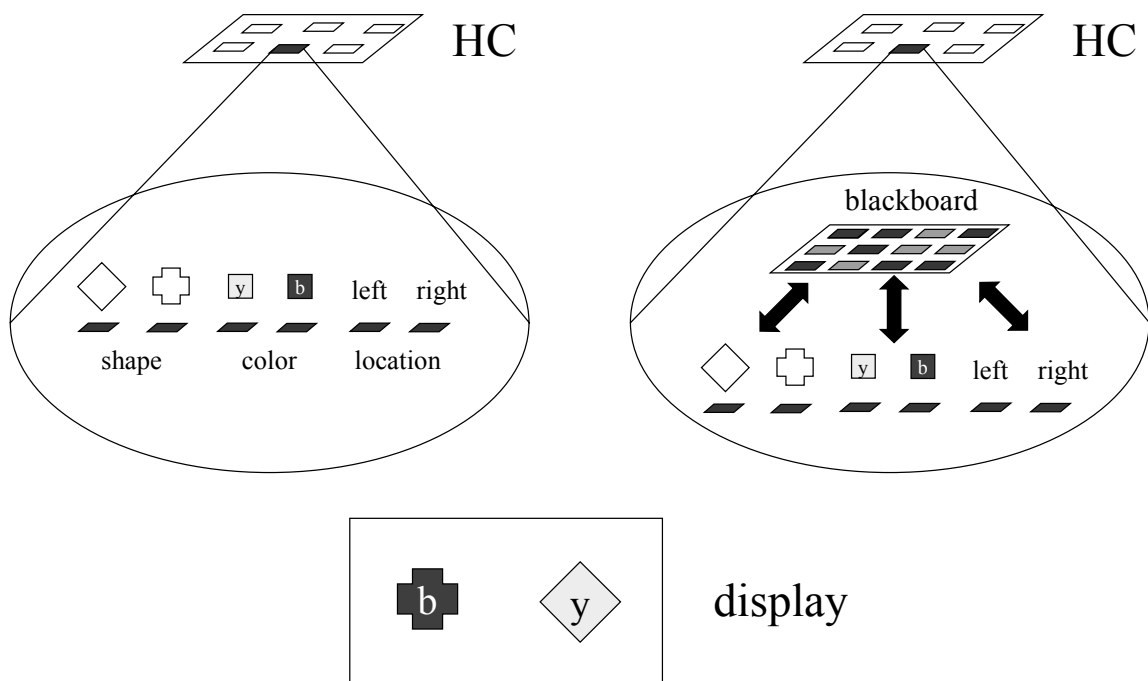


Figure 21. Left: Conjunctive encoding of the object features (e.g., shape, color, location) of two objects with a neuron (or group of neurons) in the hippocampus complex (HC). Right: Conjunctive encoding of the object features and a neural blackboard with a neuron (or group of neurons) in the hippocampus complex (HC).

Figure 21 illustrates again how a blackboard architecture can play an important role in the storage of combinatorial structures in long-term memory (i.e., in terms of synaptic modification). Even a conjunctive encoding as provided by the HC is sufficient, if the blackboard activations are included in the encoding. In fact, as in figure 9, the encoding of the blackboard alone would suffice.

Furthermore, figure 21 again illustrates the importance of using delay activity as a binding mechanism. The sustained activity in the WM blackboard provides the time for the synaptic modifications to occur. In contrast, if synchrony of activation is used to bind features in visual working memory (e.g., Luck & Vogel 1997; Raffone & Wolters 2001), it is not clear how the relations between the features can be preserved in the transition from working memory to long-term memory, that is, how information encoded with synchrony of activation can be stored in terms of synaptic modifications. If the HC forms a conjunctive encoding of the neurons that are active in a time window (event) of about 1 second (Rolls & Treves 1998), it will form a conjunctive encoding of the features of all objects in a display, in the manner illustrated in figure 21 (left). In that case, the relations between the features, expressed with synchrony of activation, are lost in the transition

from working memory to long-term memory.

7.4. Integrating combinatorial structures in language and vision

Figure 22 illustrates the combinatorial structure of *The little star is beside a big star* in terms of the architectures in figures 18 and 20. This structure can be combined with the sentence structure in figure 8. In particular, the neural assemblies for words will be connected with neurons that encode visual features or visual operations (e.g., translations, visual selections). Figure 22 illustrates in a schematic fashion how the question “Where is the little star?” can be answered in this way.

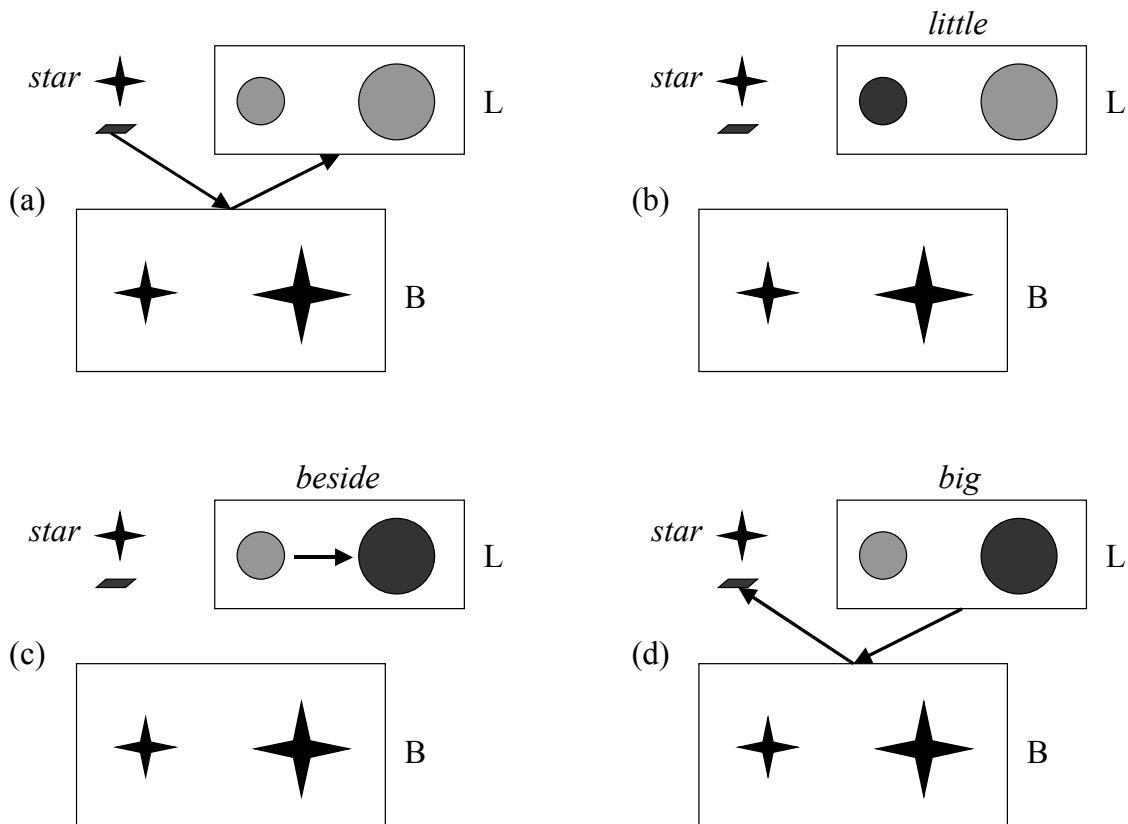


Figure 22. The structure of *The little star is beside a big star* in the neural blackboard architectures of figure 18 and 20. (a). Selection of the shape of the star (related with the word *star*) in the shape feature domain, which results in an interaction in the blackboard (B) and a selection of location and size information in the location (L) domain. (b). Selection within the location domain of the smaller size (related with the word *little*), which is now the focus of attention. (c). Shift of attention (related with the word *beside*) in the location domain to the location beside the attended location in (b). The newly selected location is now the focus of attention. (d). Feedback activation from the location domain interacts with the activation in the blackboard. This results in the selection of the feature in the shape domain (*star*, related with the word *star*) that corresponds with the location and the size (related with the word *big*) of the newly attended object in (c).

In figure 22a, the word *star* has selected the shape of the star as a cue in the shape feature domain. As a result, the cue-related activation in the feature domain is enhanced (in line with Chelazzi et al. 1993). In terms of the process illustrated in figure 19, the selection of the cue initiates an interaction in the visual blackboard (B), where the information of the visual display is processed or maintained. The interaction produces an enhancement of cue-related activation in the blackboard, which results in the selection (enhancement) of the neurons in the location domain (L) that encode the locations of the stars (Van der Velde & de Kamps 2001). The neurons in this domain also encode the spatial magnitude of the objects in a display.

In figure 22b, the word *little* initiates an enhancement (selection) of the neural activity in the location domain that encodes the location of the smaller object (which can result from selecting the neuron with the smallest receptive field that fully covers an object). The enhancement of neural activity in the location domain constitutes a form of spatial attention (Posner & Petersen 1990), so that the location of the little star is now the attended location.

In figure 22c, a shift of spatial attention is produced in the location domain. As a result, the neural activity that encodes the location of another object in the vicinity of the attended location in figure 22b is enhanced. If spatial information is (initially) encoded in eye-centered coordinates (Batista et al. 1999), a shift of spatial attention will produce a spatial transformation in terms of eye-centered coordinates. The spatial transformation involved can be used to activate the associated word assembly (*beside*).

In figure 22d, the newly selected neural activity in the location domain can be used to obtain a measure of the spatial magnitude of the newly attended object ('big', in comparison with the previous object). It can also be used to influence processing in the blackboard, so that the shape of the newly attended object can be selected in the shape feature domain. Both selections can produce the activation of their associated word assemblies (*big* and *star*).

The process illustrated in figure 22 would operate in a similar manner with a structure like *The little triangle is beside a big triangle*. The only difference would be the selection of the shape of the triangle in the shape feature domain. The processes in the other domains are not affected by the change from star to triangle. Thus, for instance, the attention shift in figure 22c operates in the same manner for the shift from a little star to a big star as for the shift from a little triangle to a big triangle, because it is only based on the information in the location domain. Likewise, in a structure like *The little diamond is above a big square*, the process in figure 22a is only affected by the (initial) selection of the shape of the diamond (instead of the star or the triangle), and the process in figure 22c is only affected by the nature of the spatial transformation (above, instead of beside).

The similarity between these examples emphasizes the combinatorial nature of the process illustrated in figure 22. Each of the individual processes operates only on information that is available in its own domain. However, by using the blackboard, a process in one domain can influence the processes in the other domains. In this way, a combinatorial structure can be produced by the architecture as a whole. For instance, with *The little diamond is above a big square*, the attention shift in figure 22c will produce the square as the second object selected in the shape feature domain (instead of the star or the triangle in the other examples), by the interaction process in the blackboard illustrated in figure 22d.

The interactions between object (feature) information and spatial information, illustrated in figure 22, have a clear relation with attentional processes in the cortex (e.g., as in the ‘bias competition’ model of attention, Desimone & Duncan 1995). The blackboard architecture in the visual cortex (figure 18) and the blackboard architecture of visual working memory (figure 20) can be combined in a ‘closed-loop attention model’ (Van der Velde et al, in press). The interaction between neural sentence structures (figure 8) and visual blackboard structures, as illustrated in figure 22, could also form the basis of a model that combines pictorial and propositional aspects of mental imagery (see Pylyshyn 2002, for a discussion on that topic).

8. Conclusion

Combinatorial structures as found in human cognition can be instantiated in terms of neural blackboard architectures. In particular, neural blackboard architectures provide a solution of the four fundamental problems that a neural instantiation of combinatorial structures is faced with (Jackendoff 2002): the massiveness of the binding problem, the problem of 2, the problem of variables and the relation between combinatorial structures in working memory and long-term memory.

Neural blackboard architectures can be formulated for sentence structure and for combinatorial structures (feature binding) in visual cognition. There are clear structural differences between these architectures, which derive from the nature of the information processing in which they are involved (in particular, the spatial arrangement of visual features in a visual display versus the sequential arrangement of words in a sentence). However, there are also important similarities between the different blackboard architectures.

One such similarity concerns the manner in which the neural blackboard architectures afford the possibility of a selective flow of activation. This selective flow of activation forms the basis of binding constituents in combinatorial structures, as exemplified in the manner in which binding questions can be answered in these architectures. Thus, the binding of features in the visual blackboard architectures consists of a selective flow of activation from one feature domain to another, determined by the interaction process in the blackboard. Likewise, the blackboard architecture for sentence structure produces a selective flow of activation in the process of answering a ‘binding’ question.

Another similarity between the architectures concerns the transition from working memory to long-term memory, and the role of delay activity as a binding mechanism. Combinatorial structures can be stored in long-term memory (using synaptic modification) when the blackboard activity is included in a conjunctive form of encoding as provided by the hippocampus complex. The delay activity in the blackboard provides the time for the synaptic modifications (Hebbian learning) to occur, even in the case of one-trial learning.

It is clear that a substantial amount of work, both theoretically and empirically, is needed for the further development of the neural basis of combinatorial structures in human cognition. However, the similarities between the neural instantiation of combinatorial sentence structure and the (more familiar) neural instantiation of combinatorial structures in visual cognition provide the hope that such a development can be successful in the near future.

NOTES

1. Tensor networks could perhaps be included here as well. However, as Fodor and McLaughlin (1990) already showed, tensor networks fail to instantiate combinatorial structures. Basically, this results from the fact that a tensor is just a list of constituents, organized in a particular fashion (i.e., as a n -dimensional list for a rank- n tensor). Any operation on a tensor consists of selecting a k -dimensional subset of the constituents in the tensor (with $k \leq n$). But all selected subsets have to be listed in the tensor beforehand, which limits the instantiation of novel structures. Furthermore, adding constituents to the tensor increases the dimensions of the tensor, which requires adjustments to all components in the cognitive system that can interact with the tensor. This limits the productivity of the cognitive system as a whole.

2. The sentence presented by Jackendoff (2002) is *The little star's beside a big star*, with the clitic z (s) to emphasize the phonological structure of the sentence. Phonological structure is not discussed here, therefore the clitic z is omitted.

3. When a symbol is copied and moved elsewhere, it is detached from its network of relations and associations. One could try to reestablish these relations and associations from time to time, but this requires an active process, executed by a control structure. Active control would be needed constantly, to decide how many of these relations and associations have to be reestablished (and how often). The frame problem in cognition could probably related with the difficulty of providing this form of control.

4. Pilot simulations showed that RNNs are very good at reproducing learned word-word associations. Thus, with the test sentence *boy hears girl*, we wanted to avoid combinations like *boy hears* and *hears girl* in the training sentences. Other than that, we wanted to train as much relations between these words as possible. In the case of this test sentence, the RNNs have learned the relation *boy Verb girl*. Furthermore, they learned the relation *dog Verb girl* with *dog hears Noun*, and the relation *boy Verb cat* with *Noun hears cat*.

5. In fact, I am not aware of pathological behavior of this kind. Broca's aphasics, for instance, often fail on sentences like *girl who dog hears obeys Mary*, but they can still understand sentences like *boy hears girl* (Grodzinsky 2000).

6. Verbs can have one, two, or three arguments, or thematic roles. Although in semantic terms many different kinds of arguments can be distinguished, they can be grouped into 'semantic macroroles' (Van Valin 2001). For simplicity, I will refer to these as 'agent', 'theme', and (later on) 'recipient'.

7. When a NP assembly is bound to a sentence structure, at least one of the memory circuits connected to the assembly is active. This activation can be used as a signal that the NP assembly is not free. Or, one could have an 'inhibition of return' that prevents the reactivation of a structure assembly that has been active recently.

8. The inhibition of the active NP assembly could result from the high transient activity that is frequently found in the cortex whenever a new stimulus appears. Due to this transient activity, the new NP assembly will win the competition before its activity reduces to a steady state. Or, the occurrence of a new noun could result in the inhibition of the active NP assembly before a new NP assembly is generated.

9. The delay assemblies do not produce inhibition directly because they consist of excitatory neurons. Instead, the delay assemblies will activate inhibitory interneurons, which in turn produce the inhibition.

10. With the question *cat chases x?*, the activation of the theme gating circuits, needed to produce the answer, should occur after the VP competition has been decided. This form of control does not depend on content information, e.g., on information about the specific VP assembly that has won the competition. Instead, it is a form of dynamic control that depends on (and regulates) the dynamics of the interaction process in the blackboard. See Van der Velde and de Kamps (2003b) for a more detailed discussion of this form of control in the blackboard architecture.

11. Although the structure in figure 9 (left) resembles those in figure 1 (bottom), they are not the same. Figure 9 (left) represents a conjunctive encoding that results from direct associations between each of the word assemblies and the HC neuron. In figure 1 (bottom), it is assumed that a specific neural circuit will activate a ‘sentence’ neuron, when the word assemblies have been activated in the correct order. Circuits of this kind are much harder to develop than the conjunctive encoding in figure 9.

12. Transitions from an attractor state into an associated attractor state have been observed in the cortex by Yakovlev et al. (1998).

13. Another motivation for a distinction between S and C assemblies is found in the formation of questions. A sentence like *The dog bites the cat* can be transformed into the question *Whom does the dog bite?* or *The dog bites whom?*. But in *The cat that the dog bites chases the mouse*, the clause *that the dog bites* cannot be transformed into the question *dog bites whom?*, to be inserted in the main sentence. Thus, the processor has to know whether *dog bites cat* is a main sentence or a relative clause.

14. The connection structure illustrated in figure 16 was not specifically designed to operate as a pushdown stack. Instead, it was designed to satisfy two constraints. First, the constraint of combinatorial productivity, which entails that every A subassembly should be able to bind with every B subassembly (and vice versa). This constraint is satisfied with the matrix-like array of columns in the connection structure. Second, the uniqueness constraint, which entails that a given A subassembly can only bind with one B subassembly (and vice versa). This constraint is satisfied with the inhibition within the horizontal and vertical rows of columns. The connection structure that results from both constraints operates as a pushdown stack if the reverberating activity in the structure decays over time.

15. A more detailed discussion of these architectures can be found in Van der Velde (1997), Van der Velde & de Kamps (2001; 2003a) and De Kamps & van der Velde (2001).

16. The image of a purple cow is used in an advertisement campaign of a brand of milk chocolate bars, which are sold in a purple wrap.

References

- Amit, D. (1989) *Modeling brain function*. Cambridge University Press.
- Amit, D. (1995) The Hebbian paradigm reintegrated: Local reverberations as internal representations. *Behavioral and Brain Sciences* 18:617-657.
- Anderson, J. R. (1983) *The architecture of cognition*. Cambridge, MA: Harvard University Press.
- Barsalou, L. W. (1999) Perceptual symbol systems. *Behavioral and Brain Sciences* 22:577-660.
- Barsalou, L. W., Simmons, W. K., Barbey, A. K. & Wilson, C. D. (2003) Grounding conceptual knowledge in modality-specific systems. *Trends in Cognitive Sciences* 7:84-91.
- Batista, A. P., Buneo, C. A. Snyder, L. H. & Andersen, R. A. (1999) Reach plans in eye-centered coordinates. *Science* 285:257-260.
- Bechtel, W. & Abrahamsen, A. (2002) *Connectionism and the mind*. Malden, MA: Blackwell.
- Bierwisch, M. (1999) Words in the brain are not just labeled concepts. *Behavioral and Brain Sciences* 22:280-282.
- Bloom, P. (2000) *How children learn the meanings of words*. Cambridge, MA: MIT Press.
- Bulter J. (2001) Feedback connections and conscious vision. *Trends in Cognitive Sciences* 5:369-370.
- Calvin, W. H. & Bickerton, D. (2000) *Lingua ex machina: Reconciling Darwin and Chomsky with the human brain*. Cambridge, MA: MIT Press.
- Caplan, D. (1992) *Language: Structure, processing and disorders*. Cambridge, MA: MIT Press.
- Chelazzi, L., Miller, E. K., Duncan, J. & Desimone, R. (1993) A neural basis for visual search in inferior temporal cortex. *Nature* 363:345-347.
- Chomsky, N. (2000) *New horizons in the study of language and mind*. Cambridge University Press.
- Christiansen, M. H., & Chater, N. (1999) Toward a connectionist model of recursion in human linguistic performance. *Cognitive Science* 23:157-205.
- Christiansen, M.H. & Chater, N. (2001) Connectionist psycholinguistics: capturing the empirical data. *Trends in Cognitive Sciences* 5:82-88.
- Churchland, P. M. (1995) *The engine of reason, the seat of the soul*. Cambridge, MA: MIT Press.
- De Kamps, M., van der Velde, F. (2001) Using a recurrent network to bind form, color and position into a unified percept. *Neurocomputing* 38-40: 523-528.
- Dennett, D. C. (1991) *Consciousness explained*. London: Allen Lane.

- Desimone, R. & Duncan, J. (1995) Neural mechanisms of selective visual attention. *Annual Review of Neuroscience* 18:193-222.
- D'Esposito, M. (2001) Functional neuroimaging of working memory. In: Handbook of functional neuroimaging of cognition, eds. R. Cabeza and A. Kingstone. Cambridge, MA: MIT Press.
- Duncan, J. (2001) An adaptive coding model of neural function in prefrontal cortex. *Nature Reviews Neuroscience* 2:820-829.
- Durstewitz, D., Seamans, J. K., & Sejnowski, T. J. (2000) Neurocomputational models of working memory. *Nature Neuroscience* 3 (sup):1184-1191.
- Elman, J. L. (1991) Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning* 7:195-225.
- Farah, M., Humphries, G. W., Rodman, H. R. (1999) Object and face recognition. In: *Fundamental neuroscience*, eds. M. J. Zigmond, F. E. Bloom, S. C. Landis, J. L. Roberts and L.R. Squire. San Diego: Academic Press.
- Felleman, D. J. & van Essen, D. C. (1991) Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex* 1:1-47.
- Fodor, J. A. & McLaughlin, B. P. (1990) Connectionism and the problem of systematicity - Why Smolensky's solution doesn't work. *Cognition* 35:183-204.
- Fodor, J. A., & Pylyshyn, Z. W. (1988) Connectionism and cognitive architecture: A critical analysis. *Cognition* 28:3-71. .
- Fuster, J. M. (1973) Unit activity in prefrontal cortex during delayed-response performance: neuronal correlates of transient memory. *Journal of Neurophysiology* 36:61-78.
- Fuster, J. M. (1995) *Memory in the cerebral cortex*. Cambridge, MA: MIT Press
- Fuster, J. M. (2001) The prefrontal cortex-An update: Time is of the essence. *Neuron* 30:319-333.
- Fuster, J. M., Bauer, R. H., & Jervey, J. P. (1985) Functional interactions between inferotemporal and prefrontal cortex in a cognitive task. *Brain Research* 330:299-307.
- Gibson, E. (1998) Linguistic complexity: Locality of syntactic dependencies. *Cognition* 68:1-76.
- Gonchar, Y. Burkhalter, A. (1999) Connectivity of GABAergic calretinin-immunoreactive neurons in rat primary visual cortex. *Cerebral Cortex* 9:683-696.
- Grodzinsky, Y. (2000) The neurology of syntax: Language use without Broca's area. *Behavioral and Brain Sciences* 23:1-71.
- Harnad, S. (1991) The symbol grounding problem. In: *Emergent computation: Self-organizing, collective, and cooperative phenomena in natural and artificial computing networks*, ed. S. Forrest. Cambridge, MA: MIT Press.
- Hawkins, J. A. (1994) *A performance theory of order and constituency*. Cambridge University Press.
- Hebb, D. O. (1949) *The organisation of behaviour*. New York: Wiley.
- Hubel, D. (1995) *Eye, brain, and vision*. New York: Scientific American Library.
- Jackendoff, R. (2002) *Foundations of language*. Oxford: Oxford University Press.
- Jackendoff, R. (in press) Précis to *Foundations of language*. *Behavioral and Brain Sciences*.

- Lewis, R. L. (1999) Accounting for the fine structure of syntactic working memory: Similarity-based interference as a unifying principle. *Behavioral and Brain Sciences* 22:105-106.
- Livingstone, M.S. & Hubel, D. H. (1988) Segregation of form, colour, movement and depth: Anatomy, physiology, and perception, *Science* 240:740-749.
- Luck, S. J. & Vogel, E. K. (1997) The capacity of visual working memory for features and conjunctions. *Nature* 390:279-281.
- Marcus, G. F. (2001) *The algebraic mind*. Cambridge, MA: MIT Press.
- McNaughton, B. L. & Nadel, L. (1990) Hebb-Marr networks and the neurobiological representation of action in space. In: *Neuroscience and connectionist theory*, ed. M. A. Gluck & D. E. Rumelhart. Hillsdale, NJ: Erlbaum.
- Miikkulainen, R. (1996) Subsymbolic case-role analysis of sentences with embedded clauses. *Cognitive Science* 20:47-73.
- Motter, B. C. (1994) Neural correlates of attentive selection for color or luminance in extrastriate area V4. *Journal of Neuroscience* 14:2178-2189.
- Nadel, L. & Moscovitch, M. (2001) The hippocampal complex and long-term memory revisited. *Trends in Cognitive Sciences* 5:228-230.
- Newell, A. (1990) *Unified theories of cognition*. Cambridge, MA: Harvard University Press.
- Oram, M. W. Perrett, D. I. (1994) Modeling visual recognition from neurobiological constraints. *Neural Networks* 7:945-972.
- O'Reilly, R. C. & Rudy, J. W. (2001) Conjunctive representations in learning and memory: Principles of cortical and hippocampal function. *Psychological Review* 108:311-345.
- Palmer-Brown, D., Tepper, J. A. & Powell, H. M. (2002) Connectionist natural language parsing. *Trends in Cognitive Sciences* 6:437-442.
- Piattelli-Palmarini, M. (2002) The barest essentials. *Nature* 416:129.
- Pinker, S. (1994). *The language Instinct*. London: Penguin.
- Pinker, S. (1998) *How the mind works*. London: Penguin
- Port, R. F. & van Gelder, T. eds. (1995) *Mind as motion: Explorations in the dynamics of cognition*. Cambridge, MA: MIT Press.
- Posner, M.I., & Petersen, S. E. (1990) The attention system of the human brain. *Annual Review of Neuroscience* 13:25-42.
- Prabhakaran, V. Narayanan, K. Zhao, Z. & Gabrieli. J. D. E. (2000) Integration of diverse information in working memory within the frontal lobe. *Nature Neuroscience* 3:85-90.
- Pullum, G. K. & Scholz, B. C. (2001) More than words. *Nature* 413:367.
- Pulvermüller, F. (1999) Words in the brain's language. *Behavioral and Brain Sciences* 22:253-336.
- Pulvermüller, F. (2000) Syntactic circuits: How does the brain create serial order in sentences?. *Brain and Language* 71:194-199.
- Pulvermüller, F., Harle, M & Hummel, F. (2001) Walking or talking? Behavioral and neurophysiological correlates of action verb processing. *Brain and Language* 78:143-168.
- Pylyshyn, Z. (2002) Mental imagery: In search of a theory. *Behavioral and Brain Sciences* 25:157-238.

- Raffone & Wolters (2001) A cortical mechanism for binding in visual working memory. *Journal of Cognitive Neuroscience* 13:766-785.
- Rainer, G. Assad, W. F. & Miller, E. K. (1998) Memory fields of neurons in the primate prefrontal cortex. *Proceedings of the National Academy of Science USA* 95: 15008-15013.
- Rao, S. C., Rainer, G. & Miller, E. K. (1997) Integration of what and where in the primate prefrontal cortex. *Science* 276:821-824.
- Rolls, E. T. & Treves, A. (1998) *Neural networks and Brain Function*. Oxford University Press.
- Ruchlin, D. S., Grafman, J., Caneron, K. & Berndt, R. S. (in press) Working memory retention systems: A state of activated long-term memory. *Brain and Behavioral Sciences*.
- Sag, I. A. & Wasow, T. (1999) *Syntactic theory*. Stanford, CA: CSLI Publications.
- Selfridge, O. G. (1959) Pandemonium: A paradigm for learning. In: *Proceedings of the symposium on the mechanisation of thought processes* eds. D. V. Blake and A. M. Uttley. London: HM Stationary Office.
- Shastri L. & Ajjanagadde, V. (1993) From simple associations to systematic reasoning: A connectionist representation of rules, variables and dynamic bindings using temporal synchrony. *Behavioral and Brain Sciences* 16:417-494.
- Van der Heijden, A. H. C., Kurvink, A. G., de Lange, L., de Leeuw, F. & van der Geest, J. N. (1996) Attending to color with proper fixation. *Perception & Psychophysic* 58:1224-1237.
- Van der Heijden A. H. C. & van der Velde, F. (1999) Cognition for selection. *Visual Cognition* 6:83-87.
- Van der Velde, F. (1995) Symbol manipulation with neural networks: Production of a context-free language using a modifiable working memory. *Connection Science* 7:247-280.
- Van der Velde, F. (1997) On the use of computation in modelling behavior. *Network: Computation in Neural Systems* 8:1-32.
- Van der Velde, F. (1999) A spy to spy on a spy: From type to token representation with cell assemblies. *Behavioral and Brain Sciences* 22:306-307.
- Van der Velde, F. & de Kamps, M. (2001) From knowing what to knowing where: Modeling object-based attention with feedback disinhibition of activation. *Journal of Cognitive Neuroscience* 13:479-491.
- Van der Velde, F. & de Kamps, M. (2002) Synchrony in the eye of the beholder: An analysis of the role of neural synchronization in cognitive processes. *Brain and Mind* 3:291-312.
- Van der Velde, F. & de Kamps, M. (2003a) A model of visual working memory in PFC. *Neurocomputing* 52-54: 419-424.
- Van der Velde, F. & de Kamps, M. (2003b) Neural assembly binding in linguistic representation. In: *European Symposium on Artificial Neural Networks*. ed. M. Verleysen. Evere: d-side publications.
- Van der Velde, F. & de Kamps, M. & Van der Voort van der Kleij, G. T. (in press) CLAM: Closed-loop attention model for visual search. *Neurocomputing*.

- Van der Velde, F., Van der Voort van der Kleij, G. T. & de Kamps, M. (2003) Lack of combinatorial productivity in language processing with Simple Recurrent Networks. Technical Report. Leiden University.
- Van der Voort van der Kleij, G. T., de Kamps, M. & van der Velde, F. (2003) A neural model of binding and capacity in visual working memory. *Lecture Notes in Computer Science*. 2714:771-778.
- Van Gelder, T. (1990) Compositionality: A connectionist variation on a classical theme. *Cognitive Science*, 14:355-384.
- Van Valin, R. D. Jr. (2001) *An introduction to syntax*. Cambridge: Cambridge University Press.
- Von der Malsburg, C. (1987) Synaptic plasticity as basis of brain organization. In: *The Neural and Molecular Bases of Learning*, eds. J. P. Changeux and M. Konishi. Chichester: John Wiley.
- Weberhuth, G. ed. (1995) *Government and binding theory and the minimalist program*. Oxford: Blackwell.
- Wilson, F.A.W., Ó Scalaidhe, S. P., & Goldman-Rakic, P. S. (1993) Dissociation of object and spatial processing domains in primate prefrontal cortex. *Science* 260:1955-1958.
- Yakovlev, V. Fusi, S., Berman, E. & Zohary, E. (1998) Inter-trial neuronal activity in inferior temporal cortex: A putative vehicle to generate long-term associations. *Nature Neuroscience* 1:310-317.