An associative cortical model of language understanding and action planning

Andreas Knoblauch, Heiner Markert, and Günther Palm

Department of Neural Information Processing, University of Ulm
Oberer Eselsberg, D-89069 Ulm, Germany
{knoblauch,markert,palm}@neuro.informatik.uni-ulm.de

Abstract. The brain representations of words and their referent actions and objects appear to be strongly coupled neuronal assemblies distributed over several cortical areas. In this work we describe the implementation of a cell assembly-based model of several visual, language, planning, and motor areas to enable a robot to understand and react to simple spoken commands. The essential idea is that different cortical areas represent different aspects of the same entity, and that the long-range cortico-cortical projections represent hetero-associative memories that translate between these aspects or representations.

1 Introduction

When words referring to actions or visual scenes are presented to humans, distributed cortical networks including areas of the motor and visual systems of the cortex become active [1, 2]. The brain correlates of words and their referent actions and objects appear to be strongly coupled neuron ensembles in specific cortical areas. Being one of the most promising theoretical frameworks for modeling and understanding the brain, the theory of cell assemblies [3, 4] suggests that entities of the outside world (and also internal states) are coded in overlapping neuronal assemblies rather than in single ("grandmother") cells, and that such cell assemblies are generated by Hebbian coincidence or correlation learning. One of our long-term goals is to build a multimodal internal representation using several cortical areas or neuronal maps, which will serve as a basis for the emergence of action semantics, and to compare simulations of these areas to physiological activation of real cortical areas.

In this work we describe a cell-assembly-based model of several visual, language, planning, and motor areas to enable a robot to understand and react to simple spoken commands [5]. The essential idea is that different cortical areas represent different aspects (and correspondingly different notions of similarity) of the same entity (e.g., visual, auditory language, semantical, syntactical, grasping related aspects of an apple) and that the (mostly bidirectional) long range cortico-cortical projections represent heteroassociative memories that translate between these aspects or representations. Fig. 1 illustrates roughly the assumed locations and connections of the cortical areas actually implemented in our model.

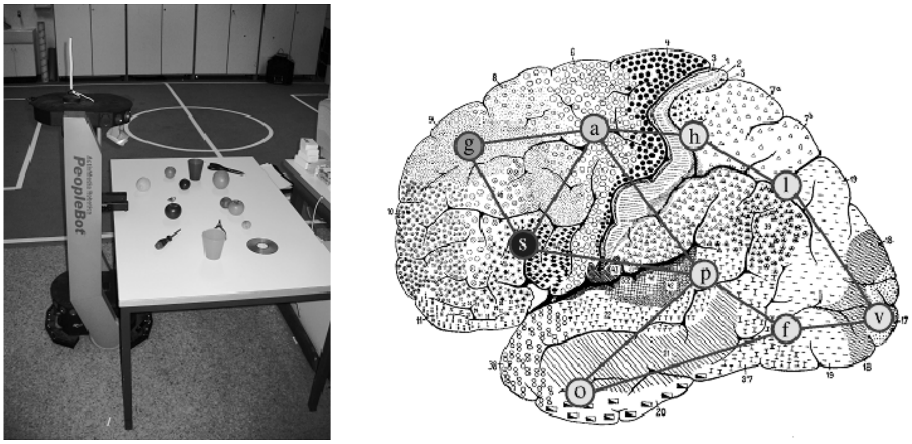

Fig. 1. Left: Robot on which the cortex model has been implemented to demonstrate a scenario involving understanding simple sentences as well as seeking and pointing to objects lying on a table. **Right:** Interaction of the different areas of the cortical model (v: visual, l: location, f: contour features, o:visual objects, h:haptic/proprioceptive, p:phonetics, s:syntactic, a:action/premotoric, g:goals/planning) and their rough localization in the human brain.

This system is used in a robotics context to enable a robot to respond to spoken commands like "bot show plum" or "bot put apple to yellow cup". The scenario for this is a robot close to one or two tables carrying certain kinds of fruit and/or other simple objects (Fig. 1). We can demonstrate part of this scenario where the task is to find certain fruits in a complex visual scene according to spoken or typed commands. This involves parsing and understanding of simple sentences, relating the nouns to concrete objects sensed by the camera, and coordinating motor output with planning and sensory processing. The cortical model system can be used to control a robot in real time because of the computational efficiency of sparse associative memories [6–8].

2 Language, finite automata, neural networks and cell assemblies

In this section we briefly review the relation between regular grammars, finite automata and neural networks [9–11]. Regular grammars can be expressed by generative rules $A \to a$ or $B \to bC$ where upper case letters are variables and lower case letters are terminal symbols from an alphabet Σ. Regular grammars are equivalent to deterministic finite automata (DFA). A DFA can be specified by a set $M = (Z, \Sigma, \delta, z_0, E)$ where Z is the set of states, Σ is the alphabet, $z_0 \in Z$ is the starting state, $E \subseteq Z$ contains the terminal states, and the function $\delta : (Z, E) \to Z$ defines the state transitions. A sentence $s = a_1 a_2 \ldots a_n \in \Sigma^*$

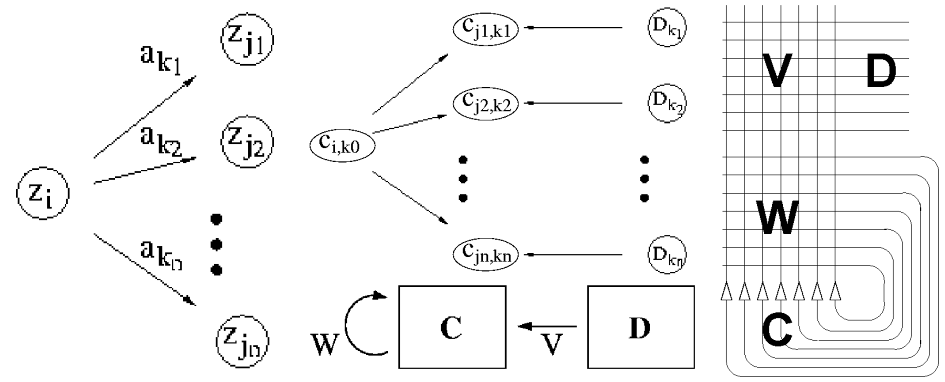

Fig. 2. Left: DFA; Middle: Neural Network; Right: Cell assemblies.

(where Σ^* is the set of all words over the alphabet Σ) is said to be well formed with respect to the grammar if $\delta(...\delta(\delta(z_0, a_1), a_2), ..., a_n) \in E$.

DFAs can be simulated by neural networks [12, 13]: E.g., it is sufficient to specify a simple model of recurrent binary neurons by $N = (C, D, W, V, c_{00})$, where C contains the local cells of the network, D is the set of external input cells, W and V are binary matrices specifying the local recurrent and the input connections (Fig. 2). The network evolves in discrete steps, where a unit is activated, $c_i(t) = 1$, if its potential $x_i(t) = (Wc(t-1) + Vd(t-1))_i$ exceeds threshold Θ_i, and deactivated, $c_i(t) = 0$, otherwise. A simple emulation of the DFA requires one neuron c_{jk} for each *combination* of state z_j and input symbol a_k, plus one neuron d_k for each input symbol a_k. Further we require synaptic connections $w_{il,jk} = d_{k,jk} = 1$ $(0 < l < |\Sigma|)$ for each state transition $(z_i, a_k) \mapsto z_j$. As threshold we use $\Theta_i = 1.5$ for all neurons. If at the beginning only a single neuron c_{0l} (e.g., $l = 0$) is active the network obviously simulates the DFA. However, such a network is biologically not very realistic since, for example, such an architecture is not robust against partial destruction and it is not clear how such a delicate architecture could be learned.

A more realistic model would interpret the nodes in Fig. 2 not as single neurons but as groups of nearby neurons which are strongly interconnected, i.e. local cell assemblies [3, 6, 14, 15]. This architecture has two additional advantages: First, it enables fault tolerance since incomplete input can be completed to the whole assembly. Second, overlaps between different assemblies can be used to express hierarchical relations between represented entities. In the following subsection we describe briefly a cortical language model based on cell assemblies.

3 Cortical language model

Our language system consists of a standard Hidden-Markov-based speech recognition system for isolated words and a cortical language processing system which

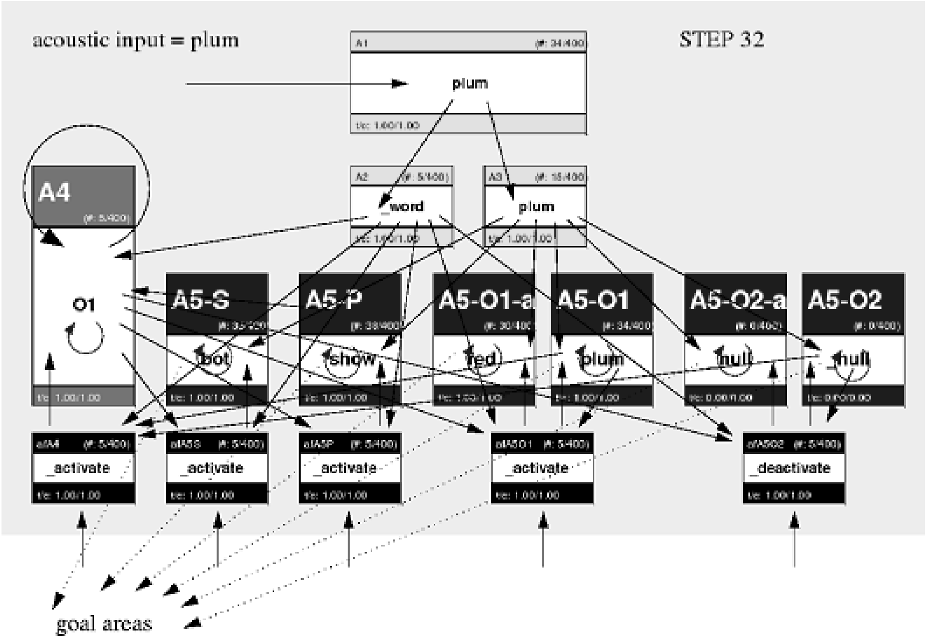

Fig. 3. Architecture of cortical language areas. The language system consists of 10 cortical areas (large boxes) and 5 thalamic activation fields (small black boxes). Black arrows correspond to interareal connections, gray arrows to short-term memory.

can analyse streams of words detected with respect to simple regular grammars [16].

Fig. 3 shows the 15 areas of the language system. Each area is modeled as a spiking associative memory of 400 neurons [17, 8]. Binary patterns constituting the neural assemblies are stored auto-associatively in the local synaptic connections by Hebbian learning.

The model can roughly be divided into three parts. (1) Primary cortical auditory areas A1, A2 and A3. (2) Grammatical areas A4, A5-S, A5-O1a, A5-O1, A5-O2a, and A5-O2. (3) Relatively primitive "activation fields" af-A4, af-A5-S, af-A5-O1, and af-A5-O2 that subserve to coordinate the activation or deactivation of the grammar areas. When processing language, first auditory input is represented in area A1 by primary linguistic features (such as phonemes), and subsequently classified with respect to function in A2 and content in A3. The main purpose of area A4 is to emulate a DFA in a similar way as the neural network in Fig. 2. Fig. 4 shows the state graph of A4. Each node corresponds to an assembly representing a grammatical state, and each edge corresponds to a state transition stored in delayed recurrent hetero-associative connections of area A4. For example, processing of a sentence "Bot show red plum" would activate the state sequence S→Pp→OA1→O1→ok_SPO corresponding to expectation of processing of a subject, a predicate, an object or attribute, and finally an object.

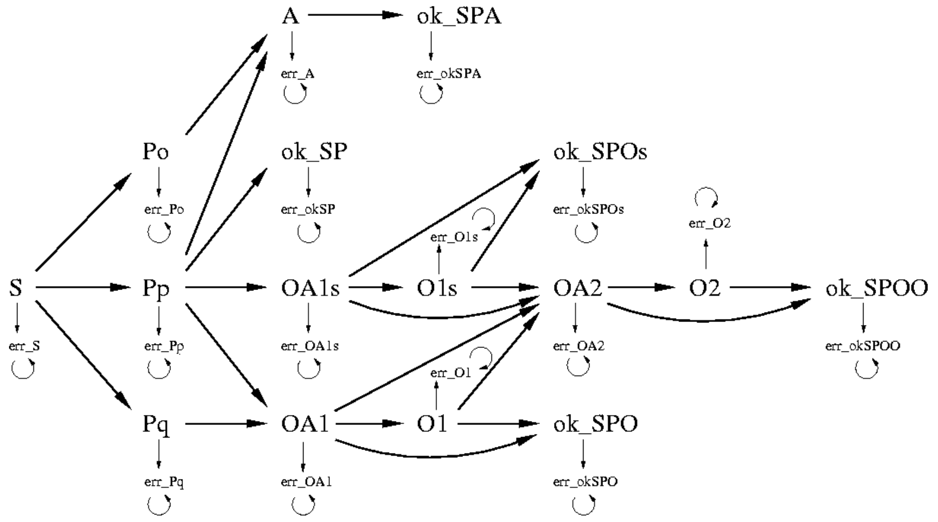

Fig. 4. Sequence assemblies stored in area A4 representing grammatical states. Each node corresponds to an assembly, each arrow to a hetero-associative link, each path to a sentence type. E.g., a sentence “Bot show red plum” would be represented by the sequence (S,Pp,OA1,O1,ok_SPO).

If the sentence was well formed with respect to the grammar, then the sequence terminates in an ”ok_X” state, otherwise in one of the ”err_X” states.

In our robot scenario it is not sufficient to decide if language input is grammatically well formed or not, but is also necessary to ”understand” the sentence by transforming the word stream into an action representation. This is the purpose of areas A5-X which correspond to different grammatical roles or categories. In our example, area A5-S represents the subject ”bot”, A5-P the predicate ”show” and A5-O1a and A5-O1 the object ”red plum”. Although establishing a precise relation to real cortical language areas of the brain is beyond the scope of this work [18, 19], we suggest that areas A1, A2, A3 can roughly be interpreted as parts of Wernicke’s area, and area A4 as a part of Broca’s area. The complex of the grammatical role areas A5 might be interpreted as parts of Broca’s or Wernicke’s area, and the activation fields as thalamic nuclei.

3.1 Example: Disambiguation using contextual information

As discussed in section 2 our associative modeling framework is closely connected to finite state machines and regular languages in that we embed the automaton states in the attractor landscape of the associative neural networks. However, in contrast to the (purely symbolic) automata our state representations can express a similarity metric (e.g., by overlaps of different cell assemblies coding different states) that can be exploited in order to implement fault tolerance against noise and to use contextual information to resolve ambiguities, e.g. by selecting the most probable interpretation.

The following example illustrates the ability of our model to resolve conflicts caused by noisy ambiguous input (see Fig. 5). After processing “bot lift” the primary auditory area A1 obtains noisy ambiguous input “ballwall” which can be interpreted either as “ball” or as “wall”. The conflict is solved by contextual information in areas A4 and A5-P representing the previously encountered verb “lift” which expects a “small” object such as “ball”, but not a large (non-liftable) object such as “wall”. Thus contextual input from area A4 (where “OAs” represents a “small” object) biases the neural activity in area A3 such that the unambiguous representation “ball” is activated.

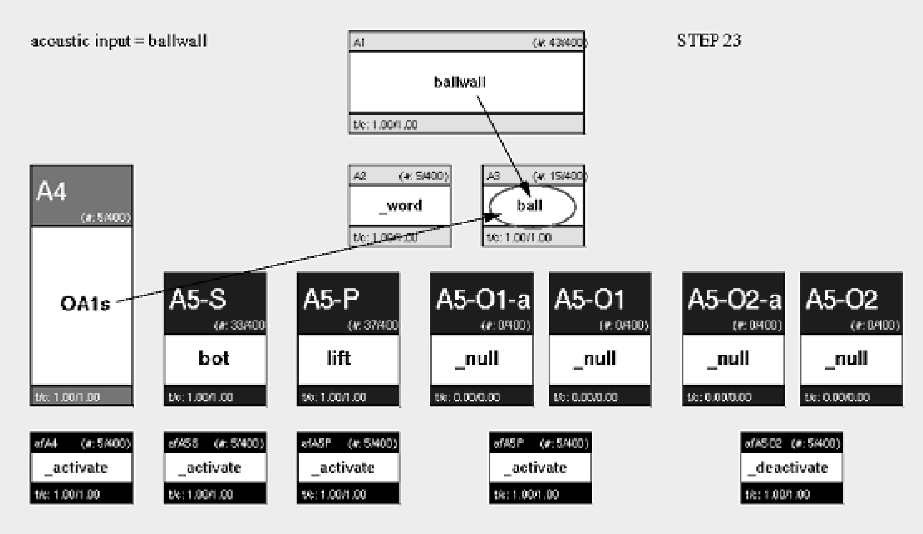

Fig. 5. Disambiguation using context: The example illustrates the states of the language areas after processing “bot lift” and then receiving noisy ambiguous acoustic input (“ballwall”) which can be interpreted either as “ball” or as “wall”. The conflict is solved by contextual information in areas A4 and A5-P representing the verb “lift” which expects a “small” object (such as a ball).

4 Action processing

Our system for cortical planning, action, and motor processing can be divided into three parts (see Fig. 6). (1) The action/planning/goal areas represent the robot’s goal after processing a spoken command. Linked by hetero-associative connections to area A5-P, area G1 contains sequence assemblies (similar to area A4) that represent a list of actions that are necessary to complete a task. For

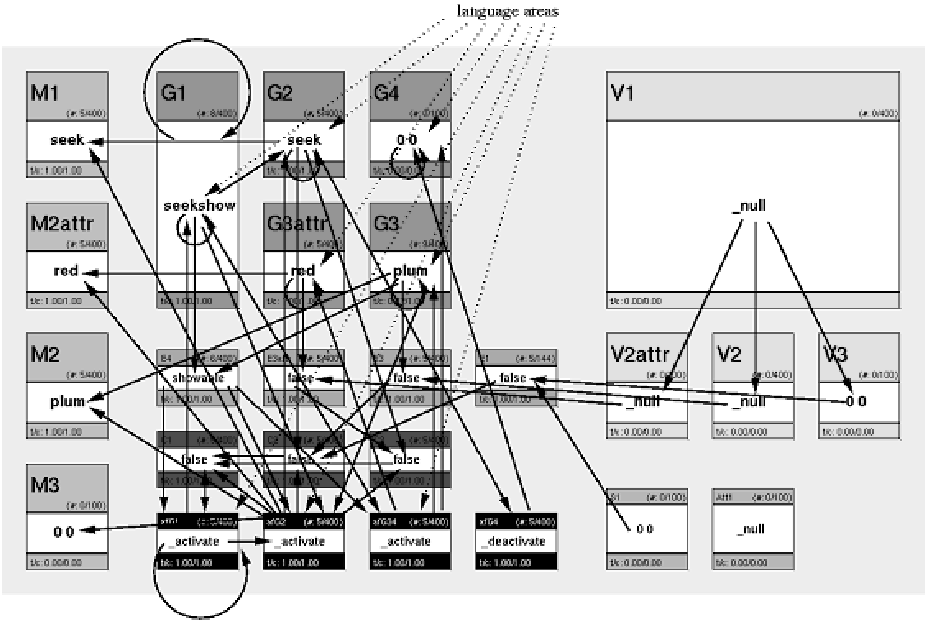

Fig. 6. Architecture of the cortical action model. Conventions as in Fig. 4.

example, responding to a spoken command “bot show plum” is represented by a sequence (seek, show), since first the robot has to seek the plum, and then the robot has to point to the plum.

Area G2 represents the current subgoal, and araes G3, G3attr, G4 represent the object involved in the action, its attributes (e.g., color), and its location, respectively. (2) The “motor” areas MX represent the motor command necessary to perform the current goal, and also control the low level attentional system. Area M1 represents the current motor action, and areas M2, M2attr, and M3 represent again the object involved in that action, its attributes, and its location. (3) Similar to the activation fields of the language areas, there are also activation fields for the goal and motor areas, and there are additional “evaluation fields” that can compare the representations of two different areas.

To illustrate how the different subsystems of our architecture work together, we describe a scenario where an instructor gives the command “Bot show red plum!”, and the robot (“Bot”) has to respond by pointing to a red plum located in the vicinity. To complete this task, the robot first has to understand the command as described in section 3, which activates the corresponding A5-representations. Activation in area A4 has followed the corresponding sequence path (see Fig. 4). Immediately after activation of the A5-representations the corresponding information is routed further to the goal areas where the first part of the sequence assembly (seekshow, pointshow) gets activated in area G1. Similarly, the information about the object is routed to areas G2, G3 and G3attr.

Since the location of the plum is unknown, there is no activation in area G3. After checking semantics, the "seek" assembly in area G2 and the corresponding representations in the motor areas MX are activated. This also triggers the attentional system which initiates the robot to seek for the plum. Fig. 7 shows the network state when the visual object recognition system has detected the red plum and the corresponding representations have been activated in areas V2, V2attr and V3. The control fields detect a match between the representations in areas V2 and G3, which initiates area G1 to switch to the next part "point" of the action sequence. The robot will then adjust its "finger position" represented in area S1 in order to point to the plum. The matching of the positions will be detected by the evaluation fields and this eventually activates the final state in G1.

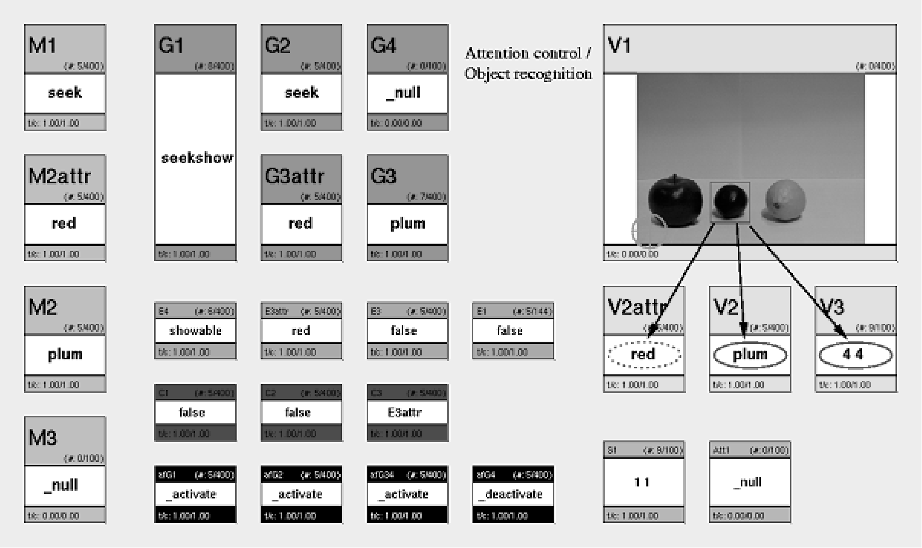

Fig. 7. State of the action planning part of the model after successfully searching for a red plum.

5 Conclusion

We have described the implementation of a cell assembly-based model of cortical language and action processing on a robot [16, 5]. The model consists of about 40 neuron populations each modelled as a spiking associative memory containing many "local" cell assemblies stored in local auto-associative connections [17]. The neuron populations can be interpreted as different cortical and subcortical areas, where it is a long term goal of this project to establish a mapping of our "areas" into real cortex [1].

Although we have currently stored only a limited number of objects and sentence types, it is well known for our model of associative memory that the number of storable items scales with $(n/\log n)^2$ for n neurons [6,7]. However, this is true only if the representations are sparse and distributed which is a design principle of our model. As any finite system, our language model can implement only regular languages, whereas human languages seem to involve context-sensitive grammars. On the other hand, also humans cannot "recognize" formally correct sentences beyond a certain level of complexity suggesting that in practical speech we use language rather "regularly".

6 Acknowledgments

We are grateful to Friedemann Pulvermüller for valuable discussions. This work has been supported by the MirrorBot project of the European Union.

References

1. Pulvermüller, F.: Words in the brain's language. Behavioral and Brain Sciences **22** (1999) 253–336
2. Pulvermüller, F.: The neuroscience of language: on brain circuits of words and serial order. Cambridge University Press, Cambridge, UK (2003)
3. Hebb, D.: The organization of behavior. A neuropsychological theory. Wiley, New York (1949)
4. Palm, G.: Cell assemblies as a guideline for brain research. Concepts in Neuroscience **1** (1990) 133–148
5. Fay, R., Kaufmann, U., Knoblauch, A., Markert, H., Palm, G.: Integrating object recognition, visual attention, language and action processing on a robot in a neurobiologically plausible associative architecture. accepted at Ulm NeuroRobotics workshop (2004)
6. Willshaw, D., Buneman, O., Longuet-Higgins, H.: Non-holographic associative memory. Nature **222** (1969) 960–962
7. Palm, G.: On associative memories. Biological Cybernetics **36** (1980) 19–31
8. Knoblauch, A.: Synchronization and pattern separation in spiking associative memory and visual cortical areas. PhD thesis, Department of Neural Information Processing, University of Ulm, Germany (2003)
9. Hopcroft, J., Ullman, J.: Formal languages and their relation to automata. Addison-Wesley (1969)
10. Chomsky, N.: Syntactic structures. Mouton, The Hague (1957)
11. Hertz, J., Krogh, A., Palmer, R.: Introduction to the theory of neural computation. Addison-Wesley, Redwood City (1991)
12. Minsky, M.: Computation: finite and infinite machines. Prentice-Hall, Englewood Cliffs, NJ (1967)
13. Horne, B., Hush, D.: Bounds on the complexity of recurrent neural networks implementations of finite state machines. Neural Networks **9(2)** (1996) 243–252
14. Palm, G.: Neural Assemblies. An Alternative Approach to Artificial Intelligence. Springer, Berlin (1982)

15. Hopfield, J.: Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Science, USA **79** (1982) 2554–2558
16. Knoblauch, A., Fay, R., Kaufmann, U., Markert, H., Palm, G.: Associating words to visually recognized objects. In Coradeschi, S., Saffiotti, A., eds.: Anchoring symbols to sensor data. Papers from the AAAI Workshop. Technical Report WS-04-03. AAAI Press, Menlo Park, California (2004) 10–16
17. Knoblauch, A., Palm, G.: Pattern separation and synchronization in spiking associative memories and visual areas. Neural Networks **14** (2001) 763–780
18. Knoblauch, A., Palm, G.: Cortical assemblies of language areas: Development of cell assembly model for Broca/Wernicke areas. Technical Report 5 (WP 5.1), MirrorBot project of the European Union IST-2001-35282, Department of Neural Information Processing, University of Ulm (2003)
19. Pulvermüller, F.: Sequence detectors as a basis of grammar in the brain. Theory in Bioscience **122** (2003) 87–103