Notice

This paper is the author's draft and has now been published officially as:

Fernando, Chrisantha (2012). Fluid Construction Grammar in the Brain. In Luc Steels (Ed.), *Computational Issues in Fluid Construction Grammar*, 312–330. Berlin: Springer.

BibTeX:

```
@incollection{fernando2012fcg,
   Author = {Fernando, Chrisantha},
   Title = {Fluid Construction Grammar in the Brain},
   Pages = {312--330},
   Editor = {Steels, Luc},
   Booktitle = {Computational Issues in {Fluid Construction Grammar}},
   Publisher = {Springer},
   Series = {Lecture Notes in Computer Science},
   Volume = {7249},
   Address = {Berlin},
   Year = {2012}}
```

Fluid Construction Grammar in the Brain

Chrisantha Fernando[1,2]

[1] School of Electronic Engineering and Computer Science, Queen Mary University of London, UK
[2] Department of Informatics, University of Sussex, Falmer, Brighton, UK

Abstract. I propose how symbols in the brain could be implemented as spatiotemporal patterns of spikes. A neuron implements a re-write rule; firing when it observes a particular symbol and writing a particular symbol back to the neuronal circuit. Then I show how an input/output function mapped by a neuron can be copied. This permits a population of neuron-based rules to evolve in the brain. We are still very far from understanding how FCG could be implemented in the brain; however, understanding how a basic physical symbol system could be instantiated is a foundation for further work.

1 Introduction

Fluid Construction Grammar is a formalism for defining and operationalizing the highly complex symbolic operations that occur in language processing [11, 53, 55, 57]. The implementations of FCG made so far are all carried out through symbolic programming languages, mostly in LISP but also in PROLOG (as discussed in other chapters of this volume [54]). How can the brain do Fluid Construction Grammar (FCG)? Constructions are rules that act on structured symbolic representations. To implement FCG the brain would need to implement a physical symbol system (PSS) [21]. Therefore my aim is to discuss the validity or otherwise of a PSS and how it can be implemented. So far I have not been able to propose any plausible neuronal mechanisms capable of the more complex matching and merging operations required by FCG. However, I am able to hypothesize neuronal implementations of symbolic re-write rules [16], to show at an algorithmic level how these rules could be evolved in the brain to develop syntactic conventions [15], and to then show at an implementation level how such rules could replicate in neuronal tissue.

My approach rests on two novelties. The first is the recent formulation of polychronous computing [31, 34], i.e. computing on the basis of spike patterns. This suggests a neural substrate for symbol structures [16]. The second is the neural replicator hypothesis proposed by Eörs Szathmáry and myself that suggests that rules operating on such symbols could be units of evolution in the brain [43]. We hypothesise that constructions of FCG replicate in the brain and evolve using a kind of neurally implemented learning classifier system [15].

Historically, purely symbolic architectures whilst in principle endlessly expressive, in practice have been hard to train, e.g. SOAR [24]. The FCG is no exception. Why is this? I suggest that it is because a grounding of FCG in lower level perceptual mechanisms is needed. This would allow local synaptic learning rules to become available to the symbolic learning system. For example, it is conceivable that a symbolic system would be able to exapt (re-use for a different function) visual and auditory shift-invariant pattern recognition mechanisms for the matching operation of FCG [37, 58]. Alternatively, it is possible that hierarchical predictive model building mechanisms originally formulated in visual perception could be re-used to construct conceptual categories [49], or that mechanisms for causal learning could be used to learn causal dependencies between symbol tokens, e.g syntactic regularities [44]. So far, such links have been poorly explored. To some extent this is because of a sociological divide between the symbolic and the connectionist factions in cognitive science [21, 23]. To help to bridge this divide it is useful to consider how chemical information is symbolic in a sense, and to realize that symbolic computation takes place in the biochemical systems of cells.

2 A Chemical Symbol System

Chemical machines, or in other words fluid automata [22] are constructed from interacting chemicals. Chemistry can be usefully thought of as containing a kind of physical symbol system. These chemical machines are very far from serial Turing machines at the implementation level, although they may well be Turing complete at the computational level [39]. An archetypical example of such a chemical machine is a cell.

What are molecules? They are objects composed of atoms that have specific structural relationships between them. A molecule is assembled according to a combinatorial syntax, i.e. a set of chemical structural constraints such as valance, charge, etc… that determine how atoms can legally join together to make the molecule. Combinatorial semantics determine how a molecule with a particular structure will react or behave in a given environment. So, semantic content in the case of the chemical symbol structure equates to chemical function, or in other words reactivity. The function of a molecule is itself a function of the semantic content of its parts, e.g. the reactivity of a benzene ring is modified by side-groups such as methyl groups. The physical symbols and their structural properties cause the system behaviour.

Note that a chemical system, whilst consisting of molecules that are symbol structures, operates in parallel (rather than in series). It is constrained by kinetic and other dynamic aspects. It is subject to non-encoded (implicit) influences such as temperature. All these aspects were not aspects which naturally came into the picture when thinking about physical symbol systems, but they do enter when considering chemical symbol systems. For good example of a symbolically specified computation in chemistry is a chemical clock. The two coupled autocatalytic cycles of the BZ reaction constitute a fluid automaton that implements a chemical clock [2, 22]. Whilst it is the symbolic (as defined above) organization of its molecules that specifies the reaction network topology, it is by the analog operation of the assembled reaction network that the clock like phenomena of the BZ reaction arises.

The properties of atoms and molecules described above give chemistry a very special set of macroscopic characteristics. For example, chemistry is **productive**. The capacity for chemical reactivity is unlimited, i.e. there are many more possible reactions than could be implemented in any realistically sized system. Indefinitely many molecules can be produced allowing indefinitely many reactions. This is possible with only a finite set of distinct atomic types. Therefore, an unbounded set of chemical structures must be composite molecules. In the same way, an indefinite number of propositions can be entertained, or sentences spoken. This is known as the productivity of thought and language, Therefore if neural symbols exist, they must have the same capacity for being combined in unlimited ways. This is not merely a crude analogy. No non-human animal has the capacity for productive thought [45].

Secondly, chemistry is **systematic**; the capacity for atoms to be combined in certain ways to produce some molecules is intrinsically connected to their ability to produce others. Consider how a chemist might learn chemistry. There are rules of thumb that help a chemist to guess how a molecule will react based on its structure. A chemist does not learn just a list of valid reactions. In the same way, there is systematicity in language, e.g. the ability to produce or understand a sentence is intrinsically connected with the ability to produce and understand other sentences. Languages aren't learned by learning a phrasebook. Languages have syntax.

Thirdly, the same atom makes approximately the same contribution to each molecule in which it occurs. For example, the contribution of hydrogen to a water molecule is to affect all sorts of properties of the reactivity of that molecule. For example, hydrogen atoms have reducing power (i.e. they suck electrons) wherever they bind in a molecule and this effect is a property of the hydrogen atom itself. This means that there is systematicity in reactivity (semantics) as well as in structure (syntax). This is known as **compositionality**. In the same way, lexical items in sentences have approximately the same contribution to each expression in which they occur. This approximate nature suggests that there is a more fundamental set of 'atoms' in language than words themselves.

Let us also consider briefly why the idea of a chemical symbol system was entertained in chemistry, that is, why scientists first came to believe in discrete atoms coming together systematically to form molecules. The crucial discoveries were of the *systematic* nature of chemistry. In Hudson's "The History of Chemistry" he describes the following discoveries [28]. Lavoisier discovered a systematic relationship in chemical reactions, i.e. the conservation of mass. Proust discovered the law of definite proportions, i.e. that compounds when broken down, produce constituents in fixed proportions. Dalton extended this to the law of multiple proportions that explained that when two elements came together to form different compounds (notably the oxides of metals), they would come together in different small integer proportions [28]. These results could elegantly be explained by an atomic theory. We see that there are analogous reasons to believe in symbols in the brain, based on an examination of the properties of human thought and language.

However, there are *extra* properties required of the PSS in cognition compared to the PSS in chemistry. Cognition includes the capacity to *learn* an appropriate PSS, not just to implement a PSS. Children can learn and manipulate explicit rules [10, 36] which implies the existence of a neural physical symbol system capable of forming structured representations and learning rules for operating on these representations [41][3].

The following is a concise definition of a symbol system adapted from Harnad to emphasize the chemical aspects [26]. A symbol system contains a set of arbitrary **atoms (or physical tokens)** that are manipulated on the basis of "**explicit rules**" that are likewise physical tokens or strings (or more complex structures, e.g. graphs or trees) consisting of such physical tokens. The explicit rules of chemistry generate reactions from the structure of atoms and molecules (plus some implicit effects, e.g. temperature). The rule-governed symbol-token manipulation is based purely on the shape of the symbol tokens (not their "meaning"), i.e., it is **purely syntactic**, and consists of "rulefully combining" and recombining symbol tokens, in chemical reactions. There are primitive atomic symbol tokens and **composite symbol-token strings (molecules)**. The entire system and all its parts – the atomic tokens, the composite tokens, the syntactic manipulations both actual and possible and the rules – are all "**semantically interpretable**:" The syntax can be systematically assigned a meaning e.g., as standing for objects or as describing states of affairs [26]. For example, semantic interpretation in chemistry means that the chemical system exhibits chemical reactivity, and in biochemistry it means that the intra-cellular chemical system stands for states of affairs in the environment outside the cell, for example the conformation of a signaling molecule may *represent* the glucose concentration outside the cell. In the same way a neural symbol system exhibits behavior such as the child's capacity to distinguish ABA from ABB in grammar learning tasks.

[3] Gary Marcus has shown that 7 month old infants can distinguish between sound patterns of the form ABA versus ABB, where A and B can consist of different sounds e.g. "foo", "baa" etc. Crucially, these children can generalize this discrimination capacity to new sounds that they have never heard before, as long as they are of the form ABA or ABB. Marcus claims that performance in this task requires that the child must extract "abstract algebra-like rules that represent relationships between placeholders (variables), such as "the first item X is the same as the third item Y", or more generally that "item I is the same as item J" [42]. Several attempts have been made to explain the performance of these children without a PSS (e.g. using connectionist models) [50] but Marcus has criticized these as smuggling in symbolic rules in one way or another by design [41, p.70]. For Marcus it seems that the system *itself* must discover the general rule. In summary, the problem with a large set of connectionist learning devices is that a regularity learned in one component of the solution representation is not applied/generalized effectively to another part [41]. Marcus calls this the problem of *training independence* [42]. Marcus considers this one of the fundamental requirements for a learning system to be described as symbolic or rule based, and I agree.

This chemical formulation may not seem of benefit, and may even be confusing to linguists, but it certainly helps me to link these two domains of computation, the biochemical and the cognitive, and this allows one to consider a new range of computations.

3 A Neural Physical Symbol System

In this section I present the outline of a neural framework for arbitrary physical tokens (atoms) arranged into molecules or symbol structures. I show how they can undergo explicit rule-governed symbol-token manipulation (reactions). Finally I show how these explicit rule sets can be learned.

In a recent paper [16] we simulated a network of cortical spiking neurons [30, 32] with synaptic weight dynamics governed by spike-time-dependent plasticity (STDP). STDP is an empirically observed process by which synaptic weights change as a function of the timing of pre- and post-synaptic spike activity. If a pre-synaptic spike reaches the post-synaptic neuron *before* the post-synaptic neuron fires, then the strength of that synapse will increase. However, if a pre-synaptic spike reaches a post-synaptic neuron *after* that post-synaptic neuron fires, then the synaptic strength will decrease. This implements a kind of causality detector. If the pre-synaptic neuron *caused* the post-synaptic neuron to fire, the synaptic strength will increase. When the extent of STDP is modulated by a reward molecule such as dopamine, it is possible to solve reinforcement learning tasks [32].

Consider first a possible neural representation of an atomic symbol token, see Figure 1. At the top we see four examples of symbol-tokens consisting of spatiotemporal patterns of spikes. The y-axis indicates which neuron the spike will stimulate, and the x-axis indicates the axons down which the spikes are passing from left to right. Thus, the depiction of the (purple) spatiotemporal pattern on the left indicates that the middle neuron is stimulated 10ms later than the top and bottom neurons (because the spikes have travelled further to the right in the top and bottom axons than the spike on the middle axon). The remainder of the figure shows the consequences of stimulating a chain of neural connections with this spike pattern in the top left box. Each chain consists of three synapses in series. There are three chains. The chain is activated by asynchronously stimulating the first three neurons on the left of the chain. That is, the top and bottom neurons are stimulated first, and then 10ms later the middle neuron is simulated. The spikes will then flow down the axons of the chain (from left to right) asynchronously activating the second and third layer neurons. It is this spatiotemporal pattern of spikes that we define as an atomic neural symbol-token. The diagram shows that detector neurons at various locations along the chain can detect this spatiotemporal spike pattern if the axonal delays from the pre-synaptic neuron to the detector neuron are properly matched to the spatiotemporal pattern such that depolarization reaches the detector neuron body simultaneously. If a summed voltage contribution from each neuron is necessary to fire the detector, then only when the appropriate spike pattern is present will

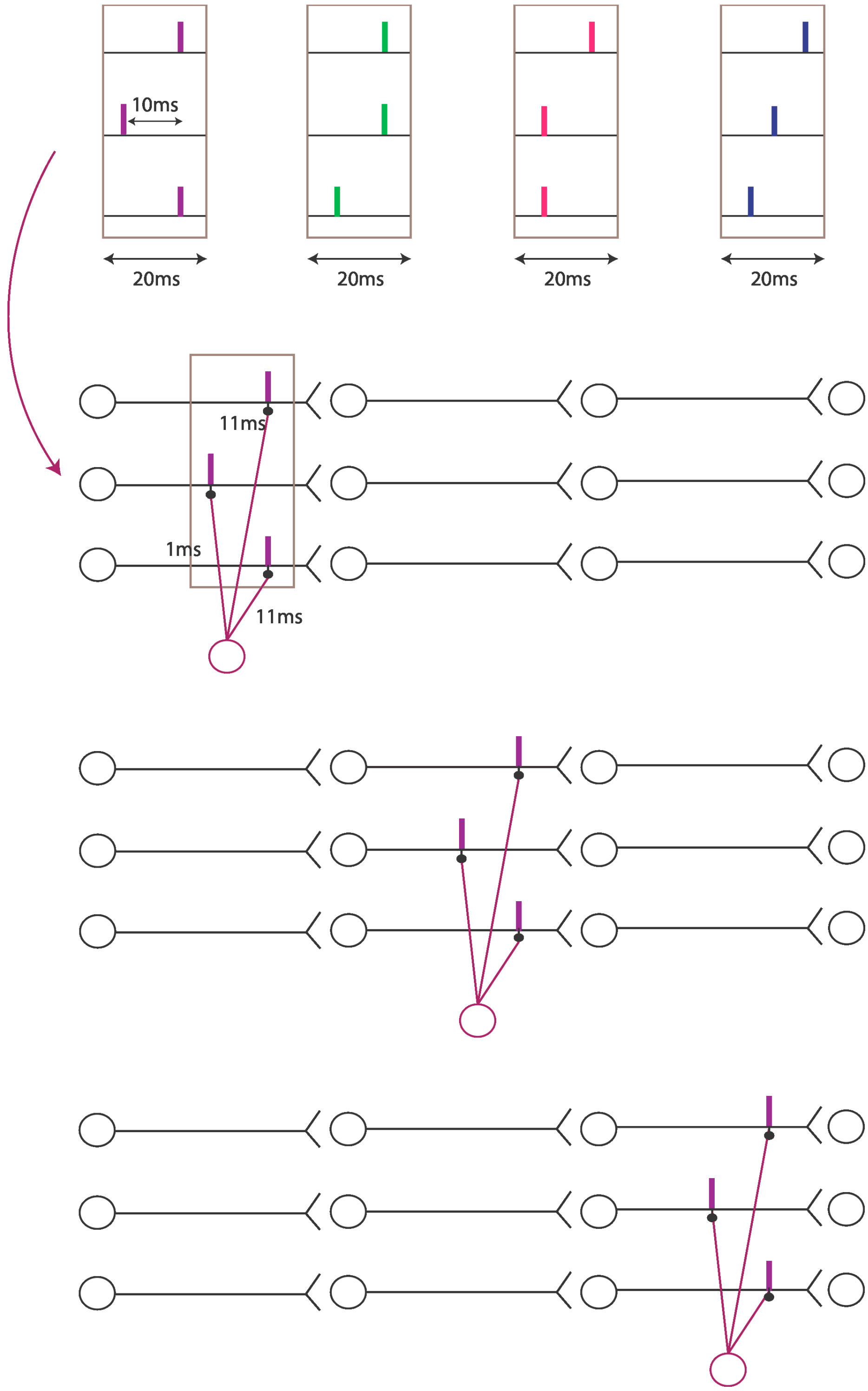

Fig. 1. Four possible spatiotemporal spike pattern based symbol-tokens are shown at the top. Below one of these spike patterns is injected into a chain of neurons running from left to right (three synaptic layers are shown). From top to bottom we see three snapshots over time as this injected symbol-tokens passes from left to right along a chain of parallel axons. Three possible detector neuron sites are shown in purple. The detector neurons inputs are arranged with a set of delays such that all three spikes reach the body of the detector neuron at the same time.

the detector fire. This implementation of neural symbol-tokens (atoms) uses the concept of polychronous computing and a modification of the concept of wavefront computing [33, 34]. Of course, in real spiking neural networks with much noise, it may be necessary to use a much larger spatial dimension in order to deal with the temporal uncertainty of the position of any one spike, and with low probability transmission at each synapse. However, the principles described here remain unchanged. Also, one should not expect the chain to be neatly visible in space. The chain is a topological concept and not a spatial concept.

The construction of molecular symbol structures from atomic symbol-tokens requires **binding** of atomic symbol-tokens together [3, 40] such that they can be subsequently manipulated (reacted) as a function of the structure of the molecule. In my framework, compositional neural symbolic structures exist as temporally ordered sequences of symbols along chains of neurons, see Figure 2.

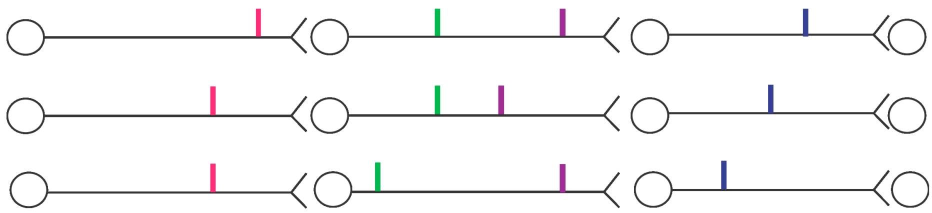

Fig. 2. A chain carrying 4 different spike patterns as a concatenated string.

This shows a snapshot of the state of a neural chain that carries the four symbol-tokens shown in Figure 1. Imagine producing this pattern by stimulating the first three neurons on the left with the blue (far right), purple, green and finally pink (far left) spike patterns in succession. Let us allocate each spatiotemporal pattern an arbitrary label, e.g. Pink (far left) = A, Green = B, Purple = C, and Blue (far right) = D for convenience. Then this symbol-structure can be described as a string or linear molecule of the form **ABCD**. I hypothesize that a great many such short chains exist in the brain. Each chain can be considered to be a kind of register in a computer, blackboard or tape that can store symbol-tokens of the appropriate size. A single symbol-token could be read by a detector neuron with the appropriate axonal delay pattern when interfacing with the chain. Similar detector neurons can exist for the symbol-tokens A, B and D and as many others as the spatial width of the chain and the temporal resolution of the neuronal detector allows.

Thus, I envisage a potentially large parallel symbol system in the brain consisting of a *population* of such chains, each capable of storing a set of symbol-token strings and operating on these strings in parallel. Interaction between (and within) such chains constitutes the operations of symbol-manipulation. Returning to the chemical metaphor, such interactions can be thought of as chemical reactions between molecules contained on separate chains, and rearrangements within a molecule expressed on the same chain. Whilst in a sense a chain can be thought of as a tape in a Turing machine (due to the serial nature of the strings), it also can be thought of as a single molecule in a chemical system (due to the existence of multiple parallel chains). This constitutes the core representational substrate on which symbol manipulation will act. The reactivity of symbol structures on these chains is described in the next Section.

A fundamental operation on a symbol token is to replace it with another symbol-token, or simply to transform it in some way, see Figure 3. The network figure shows a chain, again of three neurons width. Receiving input from the chain and writing activity back into the chain is done by a detector neuron with specific input and output delays in relation to the chain. A detector neuron (blue, bottom) only fires when the correct pattern of input is detected (as described above). In this case, the neuron's input delays are set so that it recognizes (fires for) patterns only of type D.

In the experiment the pattern of stimulation was given shown in Figure 3B. The spike raster plot and the voltage plot (Figure 3C) show two spatiotemporal patterns input to the input neurons, input pattern 1 and input pattern 2. These both fail to make the classifier neuron fire. It can be seen that in this case where the classifier fails to fire, the same pattern enters the chain as leaves the chain. This is because the spatiotemporal organization of these patterns does not match the spatiotemporal tuning curve of the detector neuron. Only when the third spatiotemporal spike pattern is input does the detector neuron fire. Once fired, the output of the detector neuron is injected back to the neurons of the chain. If the output of the detector neuron slightly precedes the normal passage of the untransformed pattern through the chain, then the refractory period of the output neurons of the chain prevents interference by the original untransformed pattern, which is thereby replaced by the new pattern specified by the detector neuron. Such a detector neuron we will now call a classifier neuron because it is a simple context free re-write rule with a condition (detection) *and* an action pole of the type seen in Learning Classifier Systems (LCS) [27].

It can be seen that such classifier neurons are selective filters, i.e. the classifier neuron is only activated if the spatiotemporal pattern is sufficiently **matched** with the axonel delays afferent upon the neuron. The above classifier implements an implicit rule. An implicit rule is a rule that operates on atomic or molecular symbol structures *without* being specified (encoded/determined/controlled) by a symbol structure itself. There is no way that a change in the symbol system, i.e. the set of symbols in the population of chains, could modify this implicit matching rule. The implicit rule is specified external to the symbol system. Whenever the symbol D passes along this chain, it will be replaced by the new symbol, irrespective of the presence of other symbols in the system.

In a symbol system (as in chemistry), symbols are manipulated (partly) on the basis of "**explicit rules**" [4]. This means that the operations or reactivity of

[4] Quoting [26, p.335]: "Wittgenstein (1953) emphasized the difference between explicit and implicit rules: It is not the same thing to 'follow' a rule (explicitly) and merely to behave 'in accordance with' a rule (implicitly). The critical difference [between an implicit and explicit rule] is in the **compositeness** (7) and **systematicity** (8) criteria.

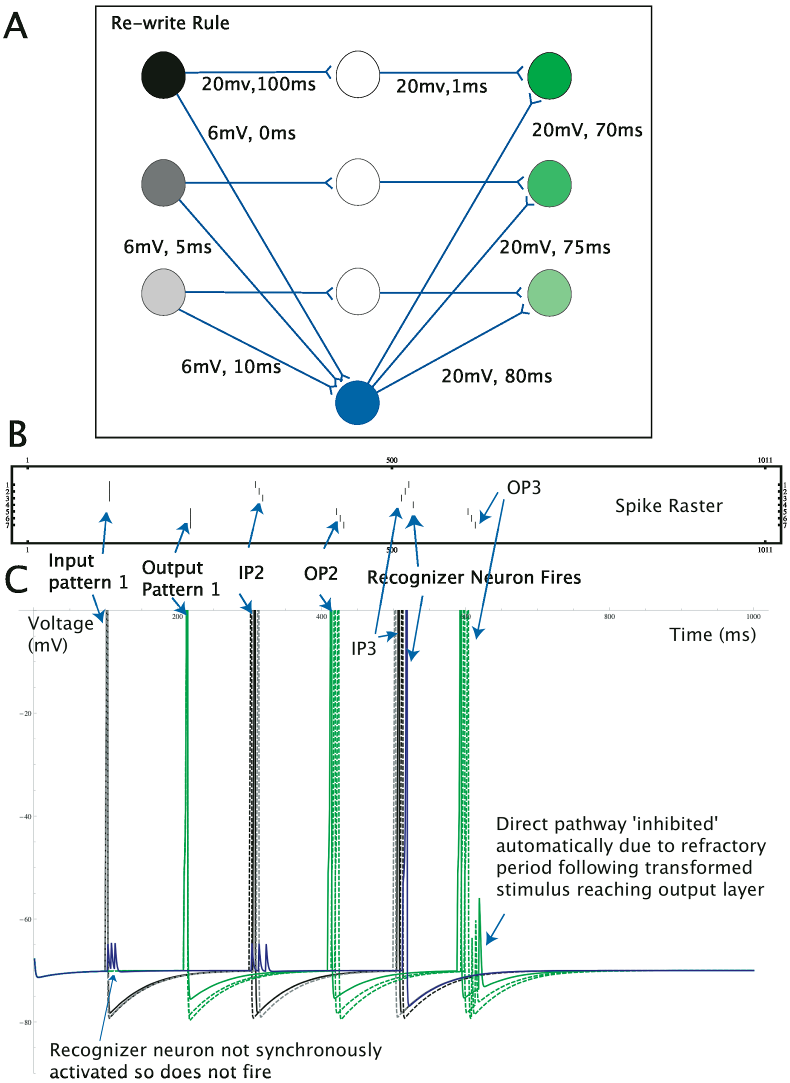

Fig. 3. The above circuit implements a context-free re-write rule. There are three input channels in this case, although it is trivial to add more. The direct pathway is by a delay line via an intermediate layer. The indirect pathway to the outputs is via a classifier neuron (blue, bottom). Only if the delays match the inter-spike interval of the input spike ensemble does the recognizer fire. Once fired, it sends signals down outputs with delays that are set so that the desired output pattern is produced. **Part B.** A spike raster showing the 3 input patterns and 3 output patterns produced in an experiment. Patterns that do not match the re-write rule pass through the classifier neuron, but those that do match the re-write rule are converted, and the passage through by the original pattern is inhibited due to the refractory period of the output neurons (see **Part C** which shows the voltages of input, output and classifier neuron). Also it is possible to explicitly inhibit the passage of the original input, but this is not needed here.

symbols depends on/is controlled by/is causally influenced by their syntactic and semantic relationship to other symbols within the symbol-structure and between symbol structures. Figure 3 above showed a classifier neuron implementing an implicit rule. This rule was not controlled by any symbols in the system; it merely operated on symbols in the system. Figure 4 below shows a classifier neuron and an inhibitory gating neuron can implement an explicit rule within our framework.

The classifier and chain shown in Figure 3 is simply modified to include an inhibitory gating unit that must receive a particular pattern of spikes (T for trigger) in order for it to become active. The simplest relation is where T immediately precedes X. Only when this is the case will the classifier neuron be disinhibited. Only when the classifier neuron is disinhibited will X be converted to Y. Otherwise X will pass through an inactive classifier (as will all other symbols). This is formally a context-sensitive re-write rule. The rule is called context sensitive because the conversion of X to Y depends on the relation of X to another contextual symbol T. A set of context-sensitive re-write rules is capable of generating a grammar of spike-patterns. Consider starting the system off with a single symbol-token S. Probabalistic application of the rules to the initial symbol S would result in the systematic production of spike patterns consisting of grammatically correct context-sensitive spike pattern based sentences. A major implementation issue in real neuronal tissue would be the fidelity of transmission of spatiotemporal spike patterns. The information capacity of such a channel may fall off with decreasing fidelity of copying in that channel in a manner analogous to Eigen's error catastrophe in genetic evolution [14].

However, the system so far described could not easily implement the kind of rule that Marcus wishes a symbol-manipulation system to learn, namely to extract "abstract algebra-like rules that represent relationships between placeholders (variables), such as 'the first item X is the same as the third item Y', or more generally that 'item I is the same as item J"' [42]. This kind of rule requires hash symbols which implement the concept of same and different, namely, If $\#_1\#\#_1$ then S, Else If $\#_2\#\#_1$ then D. That is, if the first and last string are the same, write S = same, and if the first and last strings are different write D = different. In the absence of hash symbols of this type, a classifier system would have to learn all the explicit rules for each possible pair of symbols at the first and last position, instead of learning the general rule. Both systems would be systematic, however, the system with hashes would allow a more concise specification of the same level of systematicity, and may be easier to learn. But how can such hashes

The explicitly represented symbolic rule is part of a formal system, it is decomposable (unless primitive), its application and manipulation is purely formal (syntactic, shape-dependent), and the entire system must be semantically interpretable, not just the chunk in question. An isolated ('modular') chunk cannot be symbolic; being symbolic is a systematic property... For systematicity it must be possible to combine and recombine entities rulefully into propositions that can be semantically interpreted... It is possible to devise machines whose function is the transformation of symbols, and whose operation are sensitive to the syntactical structure of the symbols that they operate upon."

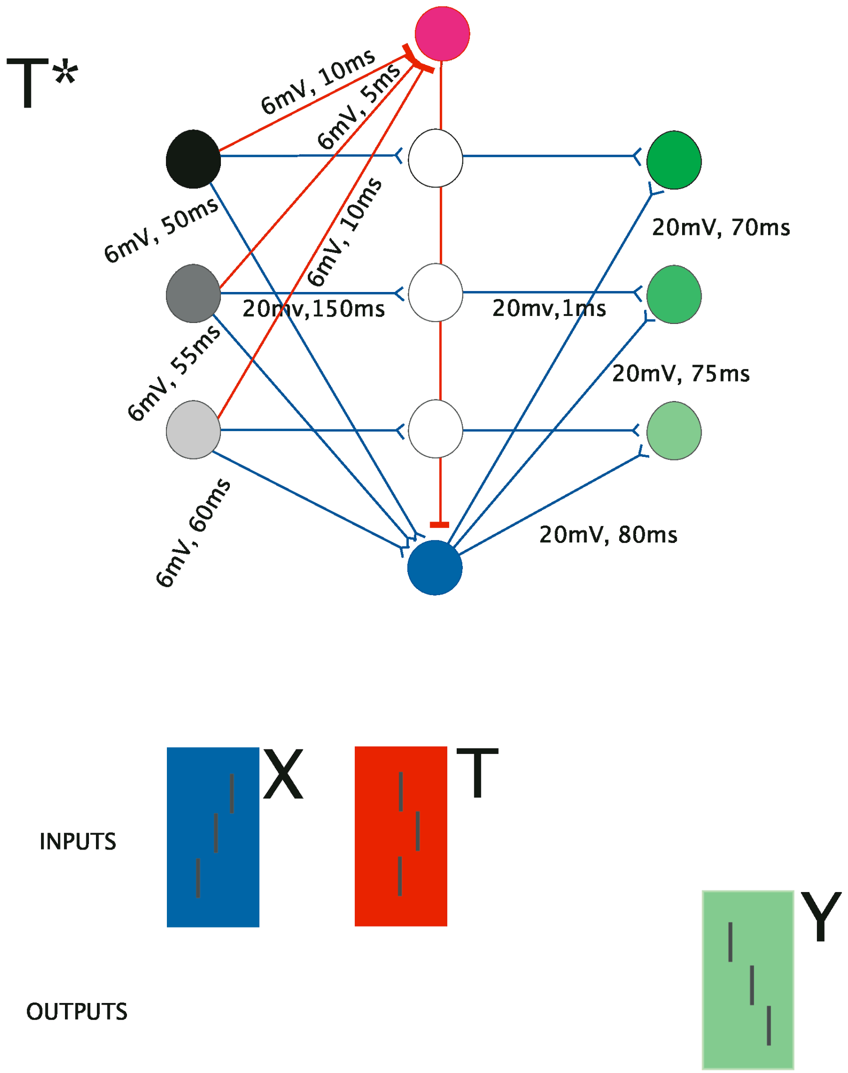

Fig. 4. An explicit rule implemented by a classifier neuron and an inhibitory gating neuron. The classifier neuron (blue, bottom) only fires if it is disinhibited by the neuron at the top (red). This occurs only if T preceeds X as these spike patterns pass down the chain from left to right. If T preceeds X, then X is converted into Y.

be implemented within my framework? One method for obtaining hashes is that a classifier neuron contains many delay lines from one channel so that it fires for a range of spike delays along that channel. Another is that it is sufficient for a classifier to be activated by only a subset of the spatiotemporal components of a symbol-token. Another possibility for implementing a same/different rule is shown in Figure 5.

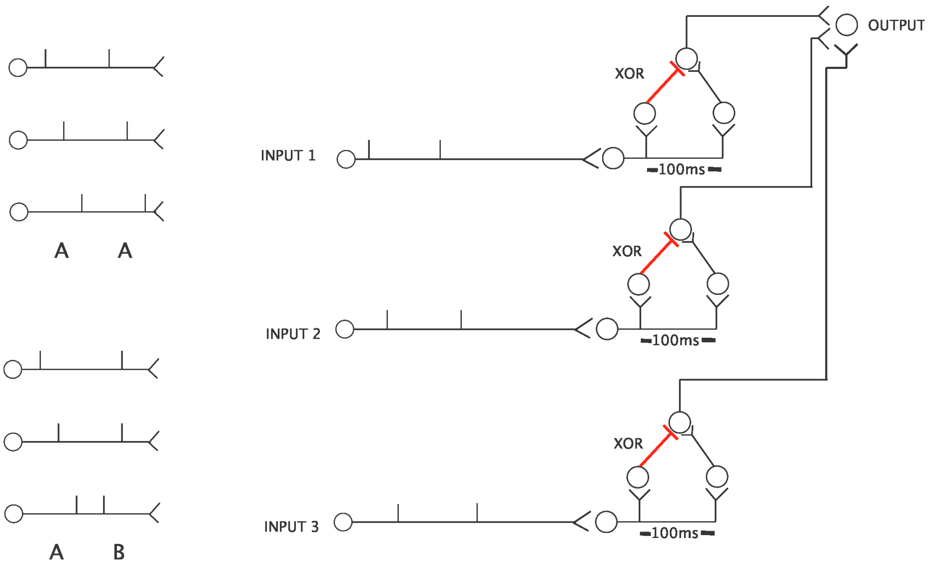

Fig. 5. A method for detecting whether successive symbol-tokens are the same or different (Left) Two pairs of sequentially presented symbols, AA and AB are shown (Right). A device that is capable of identifying consecutive symbol pairs that are different, using three XOR circuits in parallel.

On the left, the figure shows two pairs of sequentially presented symbols flowing down two reaction chains, in this case, AA on the top chain and AB on the bottom chain. On the right we see that the symbols AA from the top chain have been sent to a chain that is capable of recognizing same/different. This circuit is very simple and consists only of three XOR gates implemented by spiking neurons. The XOR function is at the heart of same/different classification because it fires 1 for the inputs 01 and 10, but fires 0 for the inputs 00 and 11. In this case, if two spikes are separated by 100ms along each channel then they will cancel each other out. However, if only one spike is present then it will be capable of activating the XOR gate. By setting the threshold of the output neuron it is possible to detect adjacent symbol tokens that differ by some specified number of spikes. The output neuron can write to the channel in the same way as described for the implicit rule action, e.g. implementing the rule, If $\#_1\#\#_1$ then S.

It seems that the neural capacity for detection of same and different is a significant departure from what can easily be achieved in chemistry! A neural physical symbol system is capable of exploiting generalization mechanisms unavailable to chemistry. In chemistry there is no known molecular mechanism by which one molecule can determine whether two other molecules are the same or different, for any more than *one* pair of such molecules. The above mechanism of detecting same and different is a neural basis for simple **matching**. We now address the more difficult question of how a symbol system can be learned, and later how hash matching can be learned.

A powerful architecture for symbolic search is XCS (accuracy based classifier system) which combines Q-learning with a population of classifiers [7]. XCS consists of a population of classifiers (which strongly resemble constructions) with condition-action poles, $C \rightarrow A$. Each classifier has a fitness F that is related to its accuracy in predicting the reward obtained in the next time step. At each point in time a subset of the classifiers (called the Match Set) will match the state of the environment. Classifiers proposing several possible actions may exist in the Match Set. An action selection method is used to select the best classifier most of the time, although sometimes actions using sub-optimal classifiers are also executed for the sake of exploration. When the action is executed and the reward obtained, then the prediction accuracy of the classifiers in the action set can be updated. Selection then takes place between classifiers in the Match Set, i.e. those with lower fitness are removed from the population. This is effectively a niche-based selection that preserves representational diversity in the population of classifiers. Learning classifier system have been used to evolve classifiers for reinforcement learning tasks such as navigation, robotic control, but also for function approximation [6] and the systematic approach used may be of interest in FCG algorithmics.

The FCG and XCS both are algorithms that require replication of classifiers (constructions). The neuronal replicator hypothesis states that replicators exist in the brain and can undergo natural selection [17–20, 56].

In order for the argument that an FCG or XCS is implemented in the brain to be plausible, and if such spatiotemporal symbols do actually exist, then it is a fundamental prior question to explain how it is possible to replicate classifiers of the type shown in Figure 3 (implicit) and Figure 4 (explicit). There are several steps to obtain replication of classifiers. The first is to understand how a single classifier can be trained. Here we return to STDP. Using the STDP based synaptic plasticity rules described previously it is possible to train a classifier neuron to fire only when exposed to a particular spatio-temporal pattern of spikes. If we wish to train the output neuron to fire only for a particular interspike interval between two input neurons, it can be done as follows. We assume that each input neuron has many pathways for communicating with the output neuron. For example dendrites form the post-synaptic neuron may connect with the axon of the pre-synaptic neuron at many locations, a not unreasonable assumption [8]. Alternatively, it may be the case that several neurons are involved in the path from input to output neuron. In the model I assume delays of 5ms, 10ms, 15ms, and 20ms each. Each weight from input to output neuron is initially sub-threshold, i.e. insufficient to allow an action potential from an input neuron to an output neuron to produce another action potential. In fact 3 input neurons must fire for the output neuron to fire. Because only two pre-synaptic neurons can contribute to a synchronous pulse, the output neuron should therefore never fire! Indeed, only if a sub-threshold depolarization is provided by an external teacher *to* the output neuron, will it fire, if at that same time it is sufficiently stimulated by pre-synaptic neurons. In our experiments, sub-threshold (training) depolarization of the post-synaptic output neuron was given 20ms after the desired condition-spike-pattern was presented to the input neurons. Due to STDP the appropriate weights from the input neurons to the output neuron increased. The tuning curve of the output neuron was entrained, confirm it was possible to train a classifier neuron to recognize particular interspike intervals [15]. The second step was to train a classifier capable of reading *and* writing a spatiotemporal spike pattern. During the training period the spike pattern to be recognized entered along the 3 input channels with spikes at 0, 50ms and 100ms latency. This pattern was presented 9 times. A short fixed time period after each input pattern was presented to the input neurons, a pattern of sub-threshold depolarization was presented to the output neurons. This output pattern was the desired output pattern, which in this case is an inversion of the original pattern (although any pattern can be trained). A set of alternative possible delay lines from each input neuron to the classifier neuron, and another alternative possible set of delay lines from the classifier neuron to each output neuron, was trained. In addition, the classifier neuron was linked to a neuromodulatory inhibitory system blocked the passage of the original spike-pattern if it was recognized. If it was not recognized then the original pattern passed through to the outputs with a delay of 120ms, unchanged in form, see [15] for a full description of the experiment.

This training procedure is sufficient for the classifier neuron to learn both the input required to activate it, and the desired output. It should be clear that the above supervised training regime for entraining the input-output function mapped by one classifier can be trivially extended to allow replication of input-output functions. This is because once a single classifier neuron has been trained, this classifier neuron can train *other* classifier neurons in the following manner. The plasticity of the first (trained) classifier neuron is held fixed. The input-spike-pattern passes now to both classifiers, and the output of the first classifier is used to produce sub-threshold output neuron depolarization in the second classifier.

Systems that are capable of being trained by supervised learning, are typically also capable of training other such systems. The mechanism of copying by supervised training/learning is exhibited in the mechanism of "didactic transfer" of receptive fields that occurs by horizontal STDP and synaptic gain modification during deafferentation of visual cortex [59]. It is also exhibited in the mechanism of copying of connection topology shown previously [18]. Recent experiments show that such temporal specific training is indeed possible [35].

4 FCG Specific Operations

Matching and merging is critical for FCG. Matching means comparison for equivalence of two symbol structures X and Y [12][9][52]. In the simplest case, X and Y are atomic symbols and there is a match if the atoms are identical. This can trivially be done by writing X and Y to a chain. They should be separated by the transformation interval; in the case of Figure 5 this is 100ms. If X and Y atoms are identical then the classifier fires. We admit the fact that this delay imposes a very severe constraint on the number of possible matches, and it is necessary to think carefully about how faster matching could be done. Let us assume that matching requires sending the two patterns to a location in the brain that can do the matching. The process by which such flexible transport can be achieved is highly non-trivial and as yet we have no explanation for this. One possibility is that matching occurs in one of the sub-cortical structures that receive many incoming connections from a wide range of cortex, e.g. the cerebellum or the basal ganglia. Indeed the striatum of the basal ganglia is responsible for matching in Anderson's ACT-R cognitive architecture, although he does not give an explanation of how it should occur there [1].

The introduction alluded to how perceptual mechanisms could be exapted for symbolic operations. An example is now given for the case of matching in FCG. The experiments in [37] use rapid reversible synaptic plasticity (dynamic link matching) to learn classes of visual transformation, e.g. reflection, rotation etc. The same mechanism can be applied to the unsupervised learning of the concept of same and different in a symbol system. The power of the method is that it can generalize, i.e. it is only necessary to show a subset of possible instances of same and different symbols for the system to be able to extend this same/different classification to novel symbols or symbol structures. The dynamic link matching algorithm has recently been applied to spiking neural networks [46]. Related algorithms are used for auditory scene analysis [5]. It is conceivable that the same perceptual mechanisms used for interpreting sensory input are also used for interpreting internally generated symbolic inputs that are similarly encoded.

A more complex case of matching occurs where X and Y are not atomic but consist of an unordered list of elements. Here X and Y are equivalent if the list contains the same elements, e.g. match('(a b c)', '(a b c)') = true but also ('(a b c)', '(b a c)') = true. The next level of matching complexity occurs when X and Y are trees. Matching can either ignore or take into consideration the order of branching, e.g. if ignored a(bc) = a(cb) but in both cases a(b(c)) != a(bc). The next step is partial tree matching, which is when some elements of X are in Y, but there are no elements in X that are not in Y: e.g. (a (d (e g))) partially matches with (a (b c) (d (e f g))).

Following matching of two symbol structures there can be merging. Merge takes already constructed objects and constructs from them a new object. Merge assumes that there has been a partial match and then adds everything of Y that is not in X to X. So when X = (a (d (e g))) partially matches with Y = (a (b c) (d (e f g))), then X becomes X' = (a (b c) (d (e f g))). Note that Y is left unchanged and can undergo further matches with other structures. The merge operation involves the copying of a symbol on the basis of the result of a match comparison. Therefore it is a type of explicit re-write rule. It is special because it requires hash based re-write, i.e. the rule does not just say if XT write TX, it says for example, if $\#_1\#_2$ write $\#_2\#_1$. That is, the re-write must work for a *range* of symbols. Whether this is plausible within our framework is not yet known. We are not yet able to provide plausible neuronal mechanisms capable of dealing with the more complex merge operations described above.

5 Discussion

There are several alternative connectionist type theories for the implementation of 'mental representations' or symbol-structures in the brain, but these are not considered in detail here [4, 38, 41, 47, 51]. I believe that it is more straightforward to face the problem head on. That is, to acknowledge that we need a full physical symbol system in the brain, and then to relax our biases about how such a physical symbol system could in fact be implemented. Thinking about a chemical symbol system helps me to do this.

There is some weak neurophysiological evidence for spatiotemporal spikes as symbol-tokens. The discovery of "cortical songs" is suggestive that discrete unique tokens such as symbols can be encoded as spatiotemporal patterns of spikes. Cortical songs are higher-order sequences of spike patterns repeated in the *same sequential order* observed in neocortical brain slices, of the form [A,C,D,E,F][A,C,D,E,F] for example where each letter represents a stereotyped polychronous pattern of activity [29]. Furthermore, there is evidence for the training methods we used to train classifiers, for example, synaptic inputs at distal dendrites can act as supervisory signals in the Hippocampus [13]. This maps to the sub-threshold depolarization we used to train classifier and output neurons. Several other papers also demonstrate methods for supervised training of spike classifiers, and so our classifier replication mechanism is by no means out of the blue. For example, the "Tempotron" is an example of learning to classify specific spatiotemporal patterns of spikes using a gradient-descent type rule to adjust weights on the basis of how rapidly a pattern results in firing of a classifier leaky-integrator neuron [25], see also [48]. Therefore, there is a growing body of work showing how replication of spatiotemporal spike pattern classifiers is possible.

In short, first I presented a plausible implementation of symbol-tokens in a brain. I then presented the core operation of an algorithm for learning symbol manipulation rules, i.e. replication of the input/output function of a classifier neuron. I have described elsewhere the details of a cognitive architecture based on a learning classifier system to learn simple syntactic rules [15]. These three components serve may provide a core for further work on understanding the neuronal basis of fluid construction grammar.

Bibliography

[1] Anderson, J.: How Can the Human Mind Occur in the Physical Universe. Oxford University Press (2007)

[2] Belousov, B.P.: A periodic reaction and its mechanism. Compilation of Abstracts on Radiation Medicine 147, 145 (1959)

[3] Biederman, I.: Recognition-by-components: A theory of human image understanding. Phychological Review 94(2), 115–147 (1987)

[4] Bienenstock, E., Geman, S.: Compositionality in Neural Systems. MIT/Bradford Books, Elsevier (1995)

[5] Bregman, A.S.: Auditory scene analysis. MIT Press, Cambridge, MA (1990)

[6] Bull, L., Kovacs, T.: Foundations of Learning Classifier Systems. Springer, Heidelberg (2005)

[7] Butz, M.: Rule-Based Evolutionary Online Learning Systems: A Principled Approach to LCS Analysis and Design, Studies in Fuzziness and Soft Computing, vol. 191. Springer, Heidelberg (2006)

[8] Chklovskii, D., Mel, B., Svoboda, K.: Cortical rewiring and information storage. Nature 431, 782–788 (2004)

[9] Ciortuz, L., Saveluc, V.: Fluid Construction Grammar and feature constraints logics. In: Steels, L. (ed.) Computational Issues in Fluid Construction Grammar. Springer Verlag, Berlin (2012)

[10] Clark, A.: In defense of explicit rules. In: William Ramsey, S.P.S..D.R. (ed.) Philosophy and Connectionist Theory. Lawrence Erlbaum (1991)

[11] De Beule, J., Steels, L.: Hierarchy in fluid construction grammar. In: Furbach, U. (ed.) Proceedings of the 28th Annual German Conference on Artificial Intelligence. Lecture Notes in Artificial Intelligence, vol. 3698, pp. 1–15. Springer Verlag, Berlin, Germany (2005)

[12] De Beule, J.: A formal deconstruction of Fluid Construction Grammar. In: Steels, L. (ed.) Computational Issues in Fluid Construction Grammar. Springer Verlag, Berlin (2012)

[13] Dudman, J., Tsay, D., Siegelbaum, S.: A role for synaptic inputs at distal dendrites: Instructive signals for hippocampal long-term plasticity. Neuron 56, 866–879 (2007)

[14] Eigen, M.: Selforganization of matter and the evolution of biological macromolecules. Naturwissenschaften 58(10), 465–523 (1971)

[15] Fernando, C.: Co-evolution of lexical and syntactic classifiers during a language game. Evolutionary Intelligence 4(3), 165–182 (2011)

[16] Fernando, C.: Symbol manipulation and rule learning in spiking neuronal networks. Journal of Theoretical Biology 275, 29–41 (2011)

[17] Fernando, C., Goldstein, R., Szathmáry, E.: The neuronal replicator hypothesis. Neural Computation 22(11), 2809âĂŞ2857 (2010)

[18] Fernando, C., Karishma, K., Szathmáry, E.: Copying and evolution of neuronal topology. PLoS ONE 3(11), e3775 (2008)

[19] Fernando, C., Szathmáry, E.: Chemical, neuronal and linguistic replicators. In: Pigliucci, M., MÃijller, G. (eds.) Towards an Extended Evolutionary Synthesis, pp. 209–249. MIT Press, Cambridge, Ma (2009)

[20] Fernando, C., Szathmáry, E.: Natural selection in the brain. In: Glatzeder, B., Goel, V., von MÃijller, A. (eds.) Toward a Theory of Thinking, pp. 291–340. Springer, Berlin (2009)
[21] Fodor, J., Pylyshyn, Z.: Connectionism and cognitive architecture: A critical analysis. Cognition 28, 3–71 (1988)
[22] Gáanti, T.: The Principles of Life. Oxford University Press, Oxford, UK (2003)
[23] Gallistel, C., King, P.: Memory and the Computational Brain: Why Cognitive Science Will Transform Neuroscience. Wiley-Blackwell (2010)
[24] Goertzel, B., Lian, R., Arel, I., de Garis, H., Chen, S.: World survey of artificial brains, part ii: Biologically inspired cognitive architectures. Neurocomputing 74, 30–49 (2010)
[25] Gutig, F., Sompolinsky, H.: The tempotron: a neuron that learns spike timing-based decisions. Nature Neuroscience 9(3), 420–428 (2006)
[26] Harnad, S.: The symbol grounding problem. Physica D 42, 335–346 (1990)
[27] Holland, J., Reitman, J.: Cognitive systems based on adaptive algorithms. ACM SIGART Bulletin 63, 43–49 (1977)
[28] Hudson, J.: The History of Chemistry. McMillan (1992)
[29] Ikegaya, Y., al, e.: Synfire chains and cortical songs: Temporal modules of cortical activity. Science 304, 559–564 (2004)
[30] Izhikevich, E.M.: Simple model of spiking neurons. IEEE Transactions on Neural Networks 14, 1539–1572 (2003)
[31] Izhikevich, E.M.: Polychronization: computation with spikes. Neural Computation 18(2), 245–82 (2006)
[32] Izhikevich, E.M.: Solving the distal reward problem through linkage of stdp and dopamine signaling. Cerebral Cortex 17, 2443–2452 (2007)
[33] Izhikevich, E.M., Gally, J.A., Edelman, G.M.: Spike-timing dynamics of neuronal groups. Cereb Cortex 14(8), 933–44 (2004)
[34] Izhikevich, E.M., Hoppensteadt, F.: Polychronous wavefront computations. International Journal of Bifurcation and Chaos 19, 1733–1739 (2009)
[35] Johnson, H., Goel, A., Buonomano, D.: Neural dynamics of in vitro cortical networks reflects experienced temporal patterns. Nature Neurosci 13(8), 917–919 (2010)
[36] Karmiloff-Smith, A.: Beyond Modularity: A Developmental Perspective on Cognitive Science. MIT Press, Cambridge, MA (1996)
[37] Konen, W., Malsburg, C.v.d.: Learning to generalize from single examples in the dynamic link architecture. Neural Computation 5(5), 719–735 (1993)
[38] Love, B.: Utilizing time: Asynchronous binding. Advances in Neural Information Processing Systems 11, 38–44 (1999)
[39] Magnasco, M.: Chemical kinetics is turing universal. Physical Review Letters 78, 1190–1193 (1997)
[40] Malsburg, C.v.d.: The what and why of binding: The modeler's perspective. Neuron 24, 95–104 (1999)
[41] Marcus, G.: The Algebraic Mind: Integrating Connectionism and Cognitive Science. MIT Press (2001)

[42] Marcus, G., Vijayan, S., Bandi Rao, S., Vishton, P.: Rule learning by seven-month-old infants. Science 283(5398), 77–80 (1999)

[43] Maynard Smith, J.: The problems of biology. Oxford University Press, Oxford, UK (1986)

[44] Nessler, B., Pfeiffer, M., Maass, W.: Bayesian computation emerges in generic cortical microcircuits through spike-timing-dependent plasticity. Theoretical Computer Science pp. 1–40 (2010)

[45] Penn, D., Holyoak, K., Povinelli, D.: Darwin's mistake: Explaining the discontinuity between human and nonhuman minds. Behavioral and Brain Sciences 31(2), 109–130 (2008)

[46] Pichevar, R., Rouat, J., Tai, L.: The oscillatory dynamic link matcher for spiking-neuron-based pattern recognition. Neurocomputing (Elsevier) 69 (2006)

[47] Pollack, J.B.: Recursive distributed representations. Artificial Intelligence 46(1), 77–105 (1990)

[48] Ponulak, F., Kasinski, A.: Supervised learning in spiking neural networks with resume: Sequence learning, classification and spike-shifting. Neural Computation (2009)

[49] Rao, R., Ballard, D.: Predictive coding in the visual cortex: A functional interpretation of some extra-classical receptive-field effects. Nature Neuroscience 2(1), 79–87 (1999)

[50] Seidenberg, M., Elman, J.: Networks are not 'hidden rules'. Trends in Cognitive Sciences 3, 288–289 (1999)

[51] Shastri, L., Ajjanagadde, V.: From simple associations to systematic reasoning: A connectionist representation of rules, variables and dynamic bindings. Behavioural and Brain Sciences 16, 417–494 (1993)

[52] Sierra, J.: A logic programming approach to parsing and production in Fluid Construction Grammar. In: Steels, L. (ed.) Computational Issues in Fluid Construction Grammar. Springer Verlag, Berlin (2012)

[53] Steels, L. (ed.): Design Patterns in Fluid Construction Grammar, Constructional Approaches to Language, vol. 11. John Benjamins, Amsterdam (2011)

[54] Steels, L. (ed.): Computational Issues in Fluid Construction Grammar, LNAI State-of-the-Art Survey, vol. 7249. Springer Verlag, Berlin (2012)

[55] Steels, L., De Beule, J., Neubauer, N.: Linking in Fluid Construction Grammars. In: Proceedings of BNAIC. pp. 11–18. Transactions of the Belgian Royal Society of Arts and Sciences, Brussels (2005)

[56] Szathmáry, E., Fernando, C.: Concluding remarks. In: Calcott, B., Sterelny, K. (eds.) The Major Transitions in Evolution Revisited, pp. 301–310. MIT Press (2011)

[57] van Trijp, R., Steels, L., Beuls, K., Wellens, P.: Fluid construction grammar: The new kid on the block. In: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics. ACL, Avignon (2012)

[58] Wiskott, L., Malsburg, C.v.d.: Recognizing faces by dynamic link matching. In: Proceedings of ICANN (1995). pp. 347–352 (1995)

[59] Young, J., Waleszczyk, W., Wang, C., Calford, M., Dreher, B., Obermayer, K.: Cortical reorganization consistent with spike timeing but not correlation-dependent plasticity. Nature Neuroscience 10(7), 887–895 (2007)