# Synfire Graphs: From Spike Patterns to Automata of Spiking Neurons

Thomas Wennekers
Department of Neural Information Processing,
University of Ulm, D-89069 Ulm, Germany
email: thomas@neuro.informatik.uni-ulm.de

**Abstract**

The concept of synfire chains has been proposed by Abeles as a reasonable biophysical model for cortical long-time correlations and replicating spike patterns in multi unit recordings. Some recent computational modelling approaches extend the model into a functional direction proposing that the synchronization of synfire chains may help to solve the binding problem of cortical information processing. In the present paper we investigate further computational aspects of synfire chains. First, we show how they can be used as spatio-temporal feature stores capable to learn, regenerate and recognize spatio-temporal signals. Thereby synfire chains introduce time into the static world of attractor neural networks as paradigms for cortical information processing. Then we extend the synfire chain model from linear autonomously evolving networks to graph-like structures with external input signals. Such *synfire graphs* can implement arbitrary deterministic and nondeterministic finite state automata. We prove formally that synfire graphs consisting of time-continuous spiking neurons can robustly process arbitrary long input words even if realistic postsynaptic potentials, bounded background noise and spike-timing jitter are taken into consideration. A single synfire node may consist of a single spiking neuron or a larger set of cells. In the latter case connections between two nodes can be diluted or have otherwise random synaptic efficacies. The extension of synfire chains to synfire graphs introduces operational (logical, procedural, cognitive) components into common modelling of Hebbian cell assemblies and brain functioning.

**Keywords:** synfire chains, synfire graphs, spike timing, synchronization, gamma-oscillations, spiking neurons, finite automata, brain theory

## 1 Introduction

The concept of synfire chains has been introduced by Abeles [1] in order to explain precisely replicating spike patterns which have been observed experimentally

in many parts of the brain [4, 31, 45, 54]. Such patterns can comprise action potentials (spikes) of different single neurons distributed over surprisingly large times of up to hundreds of milliseconds. Nonetheless within a pattern individual spikes occur with a temporal precision of about one millisecond. The classical synfire chain model proposed by Abeles provides a plausible explanation for such precise long time correlations [1, 3]: its main idea is that highly specific spatio-temporal firing patterns are generated by means of synchronously firing pools of neurons which iteratively excite well defined other pools whereby a reproducable chain of activation evolves and propagates through the network.

In sensory areas the observed patterns seem to be stimulus specific, although a single stimulus excites a collection of perhaps many different patterns [31, 54]. Moreover, experiments in frontal as well as motor areas of trained awake monkeys show that the occurrence of spike patterns in those areas is clearly correlated with behavioral events [4, 45]. These observations relate the synfire chain concept to information processing and cognitive capabilities; the explicit relation, however, is still a matter of discussion [1, 7, 56].

Reaching for a functional interpretation of cortical spike patterns and long-time correlations the present work takes the viewpoint of computer science and asks for possible computations that can be based on the synfire chain concept. It is shown that the computational capabilities of the conventional synfire chain model are much less than that of finite state automata. An extension of the standard model is proposed which at any step of a chain allows for transitions to several successor states depending on external input. Such networks are called 'synfire graphs' in the sequel. We show that synfire graph networks can simulate arbitrary finite state automata: deterministic ones (DFA) if only a single wave of activation propagates through the network at any time, and nondeterministic ones (NFA) if several chains can be present simultaneously. The presence of several activated waves can be envisaged as a generalization of Bienenstock's 'multiple synfire chains' [7].

We study synfire graph networks analytically utilizing networks of biologically plausible spiking neurons similar to integrate and fire cells. Spiking neurons mimic action potentials of real neurons in form of delta functions; they also implement realistic postsynaptic potentials and a refractory mechanism [15, 51]. First we consider networks where every synfire node consists of a single noise-free neuron. Afterwards we include bounded background noise and generalize all results to the case of pools of cells as synfire nodes with perhaps diluted or random connections between them. This way a considerable degree of realism can be reached, but nonetheless it is still possible to prove formally that synfire graph networks can properly simulate arbitrary deterministic and nondeterministic automata. In such networks a rhythmic input in the range of roughly 40Hz (for biologically plausible parameter settings) can synchronize the internal dynamics of the implemented automaton even in the presence of considerable input timing jitter and background noise.

Several earlier models for neural automata have been proposed, some of them providing Turing-complexity or even more computational power [33, 34, 38, 40, 41].

Although a high computational complexity is not the main purpose of the present paper, there still are some relations to these previous works. For example, our network architecture bears similarities to that of Omlin and Giles [41], who have shown that certain time discrete networks with sigmoid squashing functions and second order (multiplicative) synapses can stably simulate finite state automata (DFAs) for arbitrary long input words. Technically, our results present an extension of this result to continuous time networks consisting of biologically realistic spiking neurons. In the same sense synfire graph networks may provide a generalization of conventional McCulloch-Pitts networks [38].

Maass has addressed general bounds on the computational capabilities of spiking neurons [33, 34]. His results suggest that the theoretical power of spiking neurons is impressively high. However, some assumptions essential for the argumentation in [33, 34] are probably not satisfied in the brain. For example, an infinite precision in spike timing as well as the existence of perfect neural oscillators are assumed and background noise is neglected (see [8, 35, 36] for effects of noise on computing devices). These assumptions are certainly critical when imposed on real neural systems [57, 58]. In the present work we explicitely avoid them in favour of more biological realism but at the cost of reaching only less computational power. In a forthcoming paper we will show how the here proposed synfire graph networks can be integrated into a unified framework of brain functioning based on Hebbian cell assemblies [56].

# 2 Spatio-temporal spike patterns, long time correlations, and the synfire chain model

Neural agorithms in computer science are often based on discrete time networks comprising graded response neurons with sigmoid squashing (rate or output) functions [22]. Contrary, in real brains the prevailing signals communicated among neurons consist of sequences of short pulses, so-called action potentials or spikes [29]. Those in general become more frequent the higher a cell is excited. Continuous time and pulse-coding introduce an additional coding dimension in comparison with time discrete graded neurons: the rate of pulse emmision of a single neuron (or a pool of cells) can be described by the sigmoid squashing function in standard models, but spiking neurons offer the further possibility to encode information in the precise firing times of a single neuron and in timing relations between spikes of a collection of cells [1, 27, 33, 47, 52].

An increasing evidence suggests that brains in fact make use of such timing relations. Although the extent to which rate or spike coding are employed for information processing tasks in real brains is a matter of discussion [48, 58], spike timing phenomena have been demonstrated in at least two contexts:

In primary visual areas spike-*synchronization* has been observed which strongly depends on global stimulus properties like proximity of features, their colinearity,

and other simple relations between stimulus features [11, 19, 50]. It is believed that cells at distant cortical sites which contribute to the representation of the same external object express this correspondence by the synchronization of their respective firing times; cells coding for features of different objects are assumed to remain uncorrelated. This way synchronization, i.e. *short time correlations* in the range of a few milliseconds, may significiantly contribute to the problem of feature integration and segregation in preattentive visual processing (*binding by synchronization*, cf. [11, 19, 37, 50]). Comparable synchronization phenomena have also been found in other areas beside visual cortex (review in [20]). They furthermore are often accompanied by oscillations in the gamma-range (30-90Hz). Gamma-oscillations and short time synchronization of spikes have been intensively investigated theoretically [12, 15, 46, 53, 56, 58]. Many computational modeling attempts have been reviewed in [16, 57].

A second type of temporal correlations has first been described by Abeles [1, 2, 4] in higher frontal areas, but seems to be present also in other cortical and subcortical structures [31, 54, 45]. These correlations are characterized by highly precise spatio-temporal patterns in the action potential traces of one or several cells. Such patterns have been observed by Abeles in multiple unit recordings of spike trains from prefrontal areas of behaving monkeys. To this end the spike trains were searched for replicating patterns - doublets, triplets, quadruplets of spikes - with well defined delays between participating spikes (cf. Fig. 1). It turns out that some patterns occur significiantly more often than expected by chance assuming that spike trains are Poisson processes [1, 4]. These patterns can contain spikes distributed over several hundred milliseconds. Nonetheless the participating spikes are precise within roughly one millisecond. Such *long time correlations* present a puzzling problem in light of the noisy and unreliable nature of signals usually revealed by cortical recordings.

As a plausible model for the generation of spatio-temporal patterns Abeles proposed the concept of *synfire chains* [1, 3]. The main idea is depicted in Fig. 2. It assumes that patterns appear as an expression of the ordered activation of specific pools of neurons. Such pools are called synfire nodes. In cortex they are estimated to consist of roughly 30 to 100 cells ($n$ in Fig. 2) [1]. Cells in one synfire node are intensively connected to those in the next pool via diverging-converging projections, so-called synfire links. Hence, the almost synchronous firing of cells in the first pool homogeneously excites all cells in the target pool and - supposed thresholds are chosen appropriately - the cells in the target pool will also fire almost synchronously after a short transmission delay. This way a reproducable wave of activity evolves in a network of linearly ordered synfire nodes. Simultaneous recordings from two cells of the network can therefore reveal precisely correlated firing events separated by long delays, if these cells appear in distant nodes of a repeatedly activated chain.

Because it must be furthermore assumed that a single cortical neuron can belong to many different nodes which themselves contribute to different sequences, a more realistic picture of the cortical situation should be provided by the *reverberating syn-*

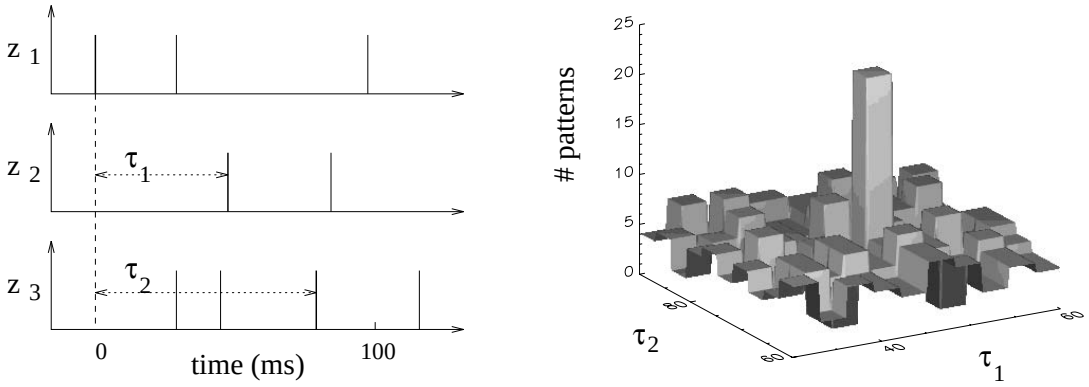


Figure 1: Triple correlation: on the left spike trains of three cells are shown. Neuron one fires at time 0; neuron two at $\tau_1 \approx 50ms$ and neuron three at $\tau_2 \approx 80ms$; these delays define a 'triplet'. The whole data set is searched for the number of occurences of all possible triplets with a spike accuracy of, say, one millisecond. Most triplets occur by chance, indicated by the homogeneous random background in the histogram on the right, but some are replicated significantly more often and appear as towers in the histogram. (Shown are computer generated data; for experimental data see [1, 3, 4]).

*fire chain model* [1]. Here, the nodes do not consist of disjoint sets of cells arranged in feedforward manner, but the neurons belonging to every single node are chosen randomly from a large collection of cells. This model is similar to hetero-associative memories storing spatio-temporal sequences of patterns (cf. next section).

Since reverberating networks have to be expected in cortex, and furthermore the probability to record from two cells belonging to the same synfire chain is only very small, most patterns observable in experiments should appear random and only a few patterns significantly above the noise level (Fig.1) [1].

Beside their puzzling physiological properties, i.e. long correlation times and precise timing, a second important observation regarding spike patterns is that their occurrence correlates with behavior [4, 45]. This has been demonstrated with trained monkeys that had to perform specific reactions in response to different stimulus situations. Although in prefrontal areas the experimental results show correlations between the occurences of patterns and external or behavioral events (such as pushing a lever) only on a coarse time scale, they nonetheless reveal that during some phases - perhaps corresponding with increased attention - almost all spikes of some cells can belong to repeating patterns. Moreover, different patterns seem to be more frequent in different stimulus situations. A similar input specifity has also been reported for repeating spike patterns in visual area V1 of monkeys by Lestienne and

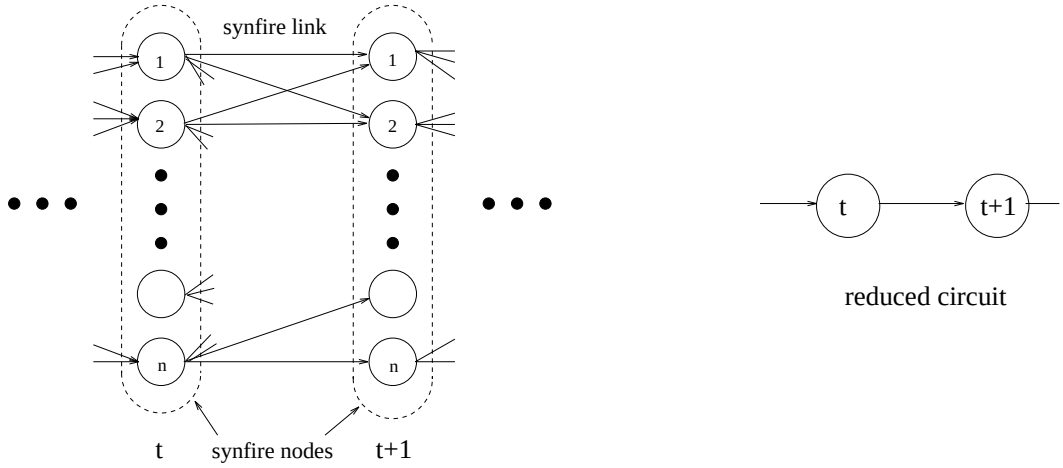


Figure 2: The synfire chain model explains precise long time correlations and cortical spike patterns: it assumes well defined distinct pools, so-called *synfire nodes* consisting of a small number of neurons (n = 30-100) which are coupled in feed-forward manner by specific *diverging/converging* projections or *synfire links*. Near synchronous firing of cells in one pool after a short transmission time evoke the synchronous activation and firing of cells in the next pool. The activity pattern in a linear sequence of pools then appears as an almost deterministic wave (*synfire chain*) and repeated waves may lead to the observation of spike patterns correlated over long times [1]. On the right of the figure an idealized scheme of the model is drawn which represents whole synfire nodes by single circles.

Strehler [31] and in the motor cortex by Riehle et al. [45].

Whereas the idea of 'binding by synchronization' provides a reasonable concept for the functional significance of gamma-oscillations and short time synchronization, such conceptual framework is almost completely missing for replicating spike-patterns and long-time correlations. In contrast to neural binding where many functional models addressing visual information processing have been investigated, by far most theoretical studies on the synfire chain model are only concerned with technical or mathematical problems.

Several early publications have addressed synfire chain-like activation in neural nets from a conceptual viewpoint [10, 21, 39, 61]. At that time, of course, evidence for the existence of such activation, for example, in form of Abeles' replicating patters, was missing. Unfortunately those early theories have not been developed very far.

More recent modeling attempts try to relate synfire chains to the neural binding-problem [3, 6, 7]. This idea has been brought up by Bienenstock. In [7] he introduced the concept of *multiple synfire chains* according to which many waves of synchronous excitation can actually be propagating in a single network at any time. Based on

multiple chains he proposed a functional model of the cortex, where complex spatio-temporal excitation structures can arise from the synchronization of simple elementary synfire chains. He argues how the concept of multiple chains can be interpreted as a basis for 'cognitive compositionality'. In some sense this work makes the above cited earlier attempts more concrete. Essentially Bienenstocks model can be viewed as a (perhaps modularized and hierarchical) associative memory which stores spatio-temporal patterns. Instead of static attractors it operates with synfire chains (= actually retrieved pattern sequences) and replaces synchronizing oscillators, as they are used in conventional models for binding, by synchronizing synfire chains.

Bienenstock denotes the complex synchronization process going on in multiple synfire chain networks as 'computation'. Although with this term he does not refer to computer science it seems interesting to investigate, which kinds of computation (now in the sense of computer science) can be performed by synfire chain networks. This is the topic of the present paper.

The rest of the paper is organized as follows. In the next section we formalize the concept of synfire chains in the framework of associative memories. Doing so, the conventional associative memory model is simultaneously extended by several properties of real neurons: our model operates in continuous time, the model neurons generate spikes as output, and the impact of spikes on target cells is described by realistic postsynaptic potentials. In section 4, we give a brief introduction into regular languages and finite automata. The main bulk of the paper is then concerned with tentative computational tasks that can be performed by synfire chains. Section 5 interprets them as spatio-temporal feature detectors, which in turn can be envisaged as recognizers for a single word of some regular language. This idea is developed further in section 6 where synfire chains are extended to synfire graphs, that is, networks of spiking neurons able to simulate arbitrary deterministic and nondeterministic finite state automata. Section 7 makes the informal arguments in section 6 more precise inasmuch as it contains proofs for the proper function of synfire graph networks comprising spiking neurons.

# 3 Synfire chains, spiking neurons, and associative memories

The reverberating synfire chain model can be reformulated within the framework of associative memories [43, 44, 63, 7]. Synfire nodes may be considered as binary patterns with $n$ ones (and zeros else) embedded in a network of $N \gg n$ cells. Synfire links between nodes correspond with Hebbian heteroassociative connections between patterns. This is made more explicit in the present section which also introduces the biologically motivated spiking neuron model employed in the sequel.

We consider a network of $N$ completely coupled neurons which operate in continuous time. At the input-side dendritic and somatic membrane properties of nerve cells are incorporated in form of 'postsynaptic potentials' (PSPs). A PSP $g_{ij}(t)$

represents the linear response kernel elicited by a Dirac-spike $z_j(t) = \delta(t)$ of cell $j$ at a postsynaptic cell $i$. We choose the prototypical form

$$g(t) \equiv g(t, \Delta, \tau) = \frac{t-\Delta}{\tau^2} \exp\left(-\frac{t-\Delta}{\tau}\right) \Theta(t-\Delta) \ , \tag{1}$$

where $\Delta \geq 0$ is a synaptic transmission time and $\tau > 0$ sets the timescale of the PSP. In general the values of $\Delta$ and $\tau$ depend on the particular synapse, that is the pair $i, j$. For simplicity we may assume that they are the same for each $i, j$. The Heaviside-function $\Theta(t)$ in (1) ensures causality of the response; $\Theta(t)$ is one for $t \geq 0$ and zero else. Furthermore the integral $\int_0^\infty g(t)dt$ is normalized to one. We describe the somatic membrane potential of neuron $i$ by a single variable $x_i(t)$ and assume linear spatio-temporal membrane properties. Then, the time-evolution of $x_i$ can be written as

$$x_i(t) = I + \frac{K_0}{n} \sum_{j=1}^{N} \int_{t_0}^{t} C_{ij} g(t-t') z_j(t') \, dt' \tag{2}$$

Here, the external input $I$ is constant and the same for all cells. The second term on the right hand side of (2) has the form of a spatio-temporal convolution elucidating the linear summation of PSPs. The inital time $t_0$ is arbitrary. $C_{ij},\ 1 \leq i, j \leq N$, are synaptic weights defined below. $K_0$ globally sets the height of PSPs and the factor $1/n$ serves for normalization purposes. $z_j(t)$ represents the temporal pattern of output spikes of neuron $j$. It consists of a sum of Dirac-functions, if spikes are modelled as Dirac-pulses. It should be noted, that for $g(t) = \delta(t-\Delta)$ the integrals in (2) can be easily performed leading to the conventional dot product form of neuronal interactions.

Firing-times at the output of a neuron have to be determined selfconsistently from equation (2): if the membrane potential $x_i(t)$ reaches a certain cell-intrinsic and time-dependent threshold $\vartheta(t)$ at time $t_{jk}$, then a Dirac-pulse is generated

$$z_j(t) = \delta(t - t_{jk}) \quad \text{if} \quad x_j(t_{jk}) = \vartheta(t_{jk} - t_{j(k-1)}) \ . \tag{3}$$

The $t_{jk},\ k = 1, 2, \cdots$ are the successive firing times of neuron $j$. After firing the threshold $\vartheta(t)$ prevents the cell from firing immediately again. This is reached by setting $\vartheta(t)$ to a large value after each spike for a fixed time $\delta$ and afterwards let it relax back to an asymptotic threshold value $\vartheta_\infty$ [6, 15, 28, 57].[1]

A set of $P \gg 1$ binary patterns $\xi^\alpha,\ \alpha = 0, 1, \cdots, P-1$, each containing $n \ll N$ ones at random positions (zero's else), is stored in the weights $C_{ij}$ according to the 'binary Hebbian rule' [44]

$$C_{ij} = \min\left(1, \sum_{\alpha=0}^{P-1} \xi_i^\beta \xi_j^\alpha\right), \quad |\xi^\alpha| = n \ll N \ . \tag{4}$$

[1] The exact implementation of refractoriness is not crucial as long as it avoids multiple spiking of neurons during the large superimposed PSP evoked by n predecessor cells in one synfire node which converge synchronously onto a given cell. Instead of internal thresholds $\vartheta(t)$ in each cell, resets of the potential $x_i$ to low values after firing as in 'integrate and fire'-neurons (see for example [51, 5, 24]) essentially lead to the same results than derived in the following.

The index $\beta$ in (4) distinguishes between two possible operational modes of the network:

If $\beta = \alpha$ the rule is called auto-associative, i.e. each pattern is associatively mapped to itself and can therefore be retrieved from noisy or incomplete input information. This corresponds with standard retrieval in attractor neural networks [26, 43, 63], although due to the time continuous nature of our model - the spikes, postsynaptic response functions and dynamical thresholds - the temporal behavior of associative memories of this kind turns out to be more complicated than that of Hopfield networks. Autoassociative networks comprising spiking neurons probably provide a good model for gamma-oscillations and short-time spike-synchronization introduced briefly in section 2. For biological (and also technical reasons, as argued soon) one should add a global threshold control that regulates the total level of activation. This control could be provided by inhibitory neurons in the cortex. For plausible time-constants the resulting associative networks oscillate in the gamma-range [12, 15, 28, 57]. Furthermore, when the stored patterns are sparse ($n \ll N$) and the overall spike activity is carefully controlled by the global inhibition, pattern retrieval can be so fast that durig each period a single pattern can be perfectly completed. Since in that case the memory capacity is also very high, we have argued that on one hand the frequency of gamma-oscillations is optimal with respect to the maximally possible repetition frequency of retrieval processes in cortex, and that on the other hand spike synchronization further supports an optimal memory capacity [57, 58].

In the later sections, we assume that the there described automata receive rhythmic synchronized input from some not further specified input network. Such input may be provided by associative memories of spiking neurons in the auto-associative mode as described here.

For $\beta = \alpha + 1$ the Hebbian learning rule (4) is 'hetero-associative'; it maps one pattern to another one. Moreover with this particular choice the patterns are stored in linear order just as suggested by the synfire chain model. For convenience it is often further assumed that the stored sequence is cyclically closed (then upper greek indices in (4) should be implicitly understood as taken modulo $P$). This way the reverberating synfire chain model is essentially recovered.

Figure 3 displays an example for the retrieval of a cyclic pattern sequence. Although individual spike trains in the lower frame (SUA, single unit activity) apparently are random due to the stochastic pattern generation, the upper plot nonetheless reveals that the network dynamics evolves deterministically. In networks of realistic size with many simultaneously stored sequencess and actually propagating waves of activity this deterministic structure might of course be hard to detect from spike recordings of just a few cells as in typical experiments. (Simulation parameters in Fig. 3 are $N = 400,\ P = 50,\ n = 10,\ \tau_{ij} = \tau = 2,\ \Delta_{ij} = \Delta = 1.45,\ \tau_{ref} = 3.,\ K_0 = 1.5,\ \vartheta_\infty = 1.0$ and $I = .8$. Only 60 of the 400 single unit spike trains are shown.)

Hetero-associative storage of spatio-temporal patterns has been repeatedly investigated in the literature, mostly for variants of time-discrete Hopfield nets (short

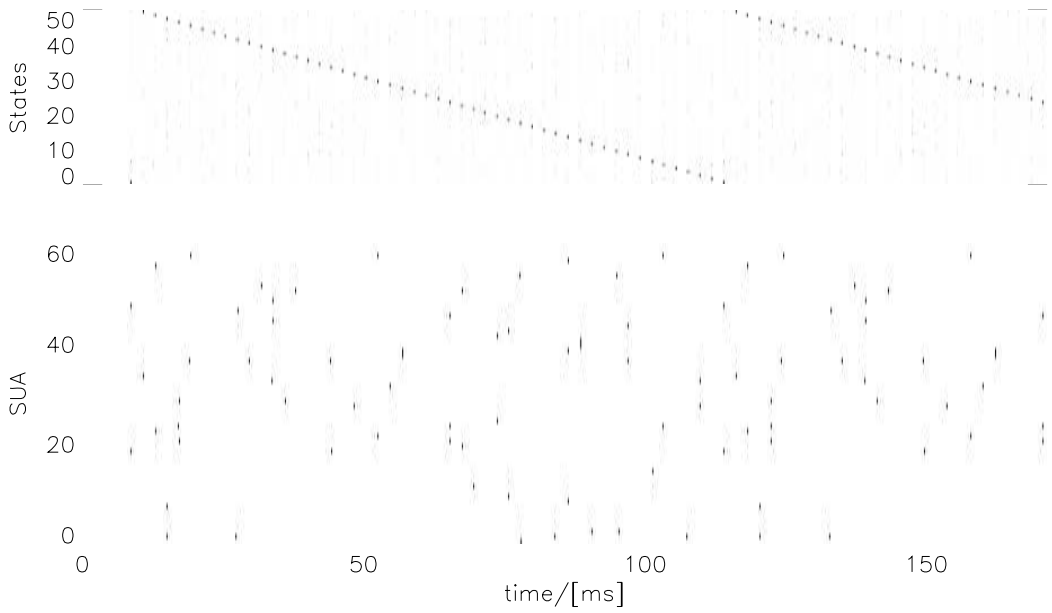


Figure 3: Retrieval of a cyclic sequence of $P = 50$ random patterns with $n = 10$ ones. Pattern zero is excited externally at time $t = 8ms$ for $.5ms$ - afterwards the network evolves autonomously. The lower frame displays single unit spikes (SUA) of 60 out of $N = 400$ cells and apparently looks random due to the stochastic nature of the patterns. The greyscale image above shows the evolution of overlaps of all 50 patterns (states) with the current network spike activity. The well organized retrieval process becomes apparent in form of dominating diagonal bands. Light grey speckled background reflects pattern overlap: cells in one pattern can also be contained in others.

review in [22]), but also for spiking neurons [1, 6, 17, 24, 56, 60]. Theoretical studies regarding synfire chains addressed their general dynamical properties [1, 5, 9, 24, 55, 60], memory capacities [7, 24], the propagation and synchronization of several simultaneously excited chains in a single network [6, 7, 3], learning or self-organization of synfire chains in networks of spiking neurons [23], and propagation velocities of activity peaks in a linear chain [6, 55].

# 4 Finite automata and formal languages

*Regular languages* are the smallest class within the Chomsky hiearchy of formal languages [25]. They are generated by regular grammars. A *regular grammar* is formally defined as a quadruple $G = (V, \Sigma, P, S)$. V is a finite set of syntactic variables or nonterminal symbols. $\Sigma$ is the *alphabet*, e.g. a finite set of (terminal) symbols with $V \cap \Sigma = \emptyset$. P is a finite set of production rules of the form $A \to a$

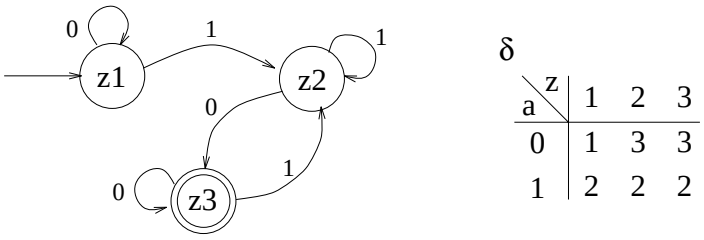


Figure 4: Simple DFA with 3 states and transition function $\delta$

or $A \to aB$ with $A, B \in V$ and $a \in \Sigma$. And, finally, $S$ is a designated element of V, the *start variable*. Successive application of rules in P starting from S eventually leads to strings or *words* consisting of terminal symbols. The set of all such possible derivations is, by definition, the regular language $L(G)$ generated by G.

Regular languages correspond with *deterministic finite automata (DFAs)*. A DFA is a quintuple $M = (Z, \Sigma, \delta, z_0, E)$. Here, $Z = \{z_1, \cdots, z_n\},\ n < \infty$ is the *set of states* and $\Sigma = \{a_1, \cdots, a_m\},\ m < \infty$ the *alphabet*. $z_0 \in Z$ is the *initial state* and $E \subseteq Z$ the set of *final or accepting states*. Operation of the automaton is defined by the *transition function* $\delta : Z \times \Sigma \to Z$ which maps every pair $(z, a) \in Z \times \Sigma$ to exactly one successor state. A *word* (over $\Sigma$) is any finite string consisting of symbols in $\Sigma$. A word x is said to be accepted by a DFA M, if reading the word symbol by symbol and starting from the initial state, application of the respective transition rules leads to a final state in E when the word is read completely. The set of words $L(M)$ that a DFA M accepts is the language M recognizes. Alternatively M can also be interpreted as a generator of $L(M)$ if starting from S possible rules are applied successively and the symbols in $\Sigma$ envisaged as output symbols. It is known that the set of languages which can be recognized by deterministic finite automata are exactly the regular languages.

A simple example for a deterministic finite automaton is shown in Fig. 4. Here, the states are $Z = \{z_1, z_2, z_3\}$, $z_1$ is the inital state S, and $E = \{z_3\}$ is the set of accepting states. As usual S is marked by an ingoing arrow in Fig. 4 and the accepting state(s) are indicated by double circles. The input alphabet of the shown automaton consists of the symbols 0 and 1, e.g. $\Sigma = \{0, 1\}$. On the right in Fig. 4 the transition function $\delta$ is given. Each pair $(z_i, a_j) \in Z \times \Sigma$ is mapped to a successor state $z_k \in Z$ and these transitions can be identified in the graphical representation of the automaton on the left by directed edges from $z_i$ to $z_k$ labeled by the respective input symbol $a_j$. The displayed automaton recognizes regular expressions of the form $0^*(1^+0^+)^+$ where a $+$ sign at a symbol or subexpression means one or more repetitions, and a $*$ indicates an arbitrary number of repetitions including zero. Hence, the shortest words that the automaton recognizes are $10, 010, 0010, 0110, 0100, 1100, 1010, \cdots$, and so on.

Beside deterministic finite automatons also *nondeterministic finite automatons*

*(NFAs)* can be defined. Those are represented by a quintuple $M = (Z, \Sigma, \delta, S, E)$, where Z,$\Sigma$ and E are defined as before. In contrast to the deterministic case in every time step several states of the NFA can be active: first, S is now a (non empty) subset of Z, e.g. initially more than just a single state can be active. Second, from every state several (including zero) successor states are possible for the same input symbol. Mathematically this is expressed by the transition function $\delta$ which now maps from $Z \times \Sigma$ into the power set of states $\mathcal{P}(Z)$. A word $x$ is accepted by an NFA if any sequence of states exists which, starting from an initial state, ends in an accepting state while x is read in completely. This sequence must not be unique. The set of words accepted by an NFA is the language it recognizes. It is known that NFAs, just as DFAs, recognize exactly the regular languages. Thus NFAs do not have more computational power. Nonetheless, they can be faster inasmuch as several state transitions can be performed on several activated states in a single step. This provides some simple kind of 'parallel processing'.

# 5 Synfire chains as spatio-temporal feature detectors

As outlined in section 3, synfire chains can be interpreted as an extension of the standard associative memory from static or structural patterns to spatio-temporal ones (see also [42] chapter 11). The example in this section is a consequent elaboration of this. In fact, the regeneration of ordered sequences of patterns has been demonstrated in SFC-models earlier [3, 5, 6, 7]. Nonetheless in this section we go a step farther and show that SFCs cannot only be used to recover sequences, but also to learn time-patterns and recognize them in a fault-tolerant manner.

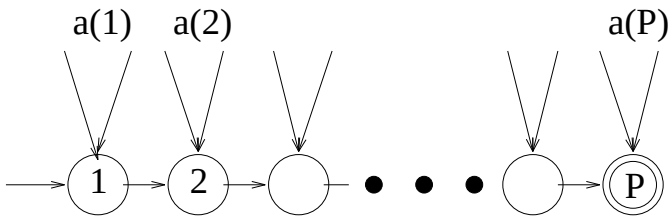


Figure 5: Scheme of a synfire chain as a spatio-temporal pattern recognizer or generator. The cells 1 to $n$ represent a conventional synfire chain. These cells receive a further input $a(t)$ which has to be learned, recognized or reproduced.

We demonstrate the main ideas in form of an example. To this end imagine an excitatorily connected associative network of spiking neurons as considered before. Inhibitory interneurons do no harm as long as the inhibition is not too strong to forbid the stable propagation of synfire activity. Assume that a sequence of P

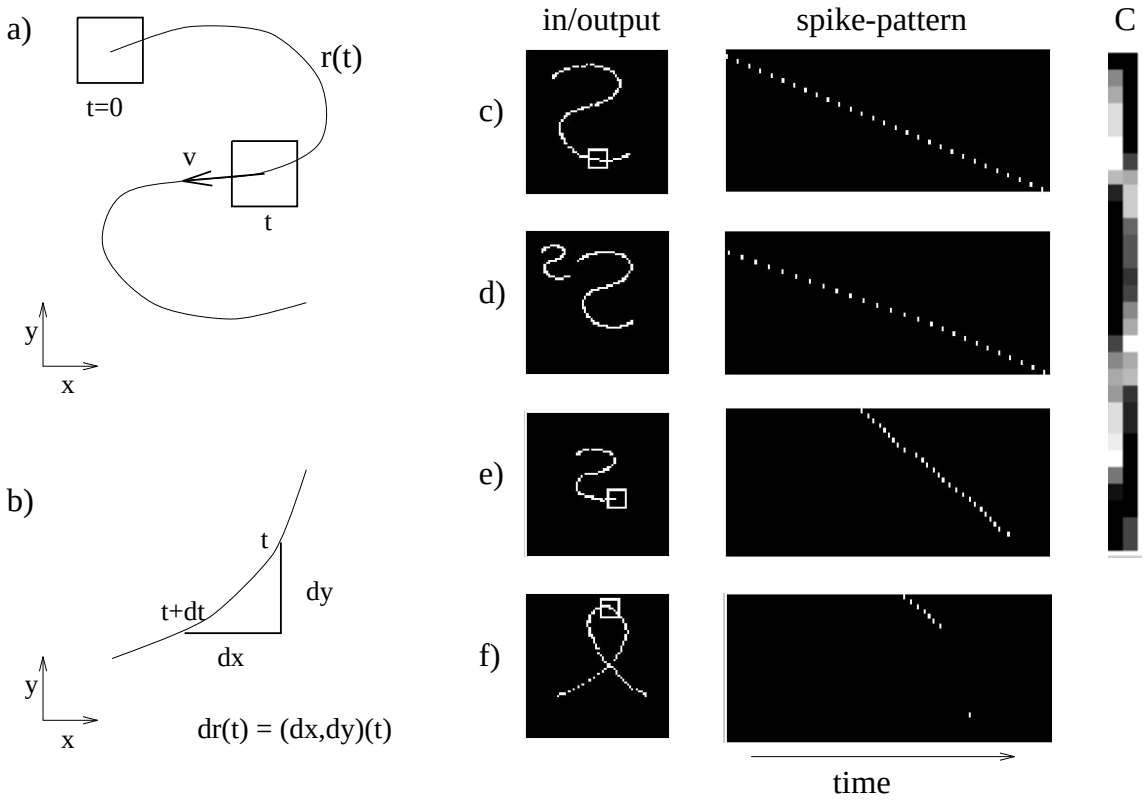


Figure 6: Synfire chains can store, recognize and replay spatio-temporal patterns. a)&b) Time patterns to learn are derived from simple line-drawings: in a) a spot moves along the curve r(t) with constant velocity; the two components of the velocity vector - proportional to dr(t) in b) - provide inputs to each neuron in a synfire chain (SFC). c) displays learning of a trajectory: to the left a snapshot of the input space is shown and to the right the spike-raster of the synfire chain neurons. Learning in c) is Hebbian: movement (the spot) and SFC both start at t=0. Each time a SFC-neuron fires it stores the actual input values dr(t) in its synapses. The finally learned synaptic input matrix $C$ is shown to the right. d) displays two-fold replay of the trajectory by repeated activation of the SFC with two different gain values. Here, the previously learned synapses are used in reverse direction and control the movement (velocity vector) in input space. e) Recognition of a distorted input trajectory. f) Recognition fails for a different pattern. For explanations see text.

patterns is stored in linear order in the coupling matrix. For sake of simplicity, we further replace each pattern by a single representative cell; hence, the synfire chain network can be thought of as consisting of P cells coupled feedforward in linear order. Some mathematical analysis reveals, that this structure can show stable propagation of activity moving from the first to the last neuron in the chain, provided the global input $I$ is chosen appropriately [55, 60]. Now, we want to learn, recognize and replay a certain time-pattern $a(t)$, where $a(t)$ may have more than one, for example $m$, components. To do this, we assume that synaptic connections $C_{ij},\ i = 1, 2, \cdots, P,\ j = 1, \cdots, m$ exist from each component of $a$ to each neuron in the SFC. These synapses are used to store samples of the pattern $a(t)$ at certain times provided by the ordered firing of the SFC neurons (Fig. 5). Learning can proceed in a Hebbian way: suppose that by some mechanism the learning pattern $a(t)$ and the activation of the SFC (node 1) start simultaneously. Then, each time $t_i$, a neuron of the SFC fires, it suffices to store the actual values $a(t_i)$ in the synapses $C_{ij},\ j = 1 \cdots m(= 2)$ converging to the respective neuron $i$. This works in a single trial ('one-shot' learning).

Figure 6c displays an example simulation (with $P = 30$ and $m = 2$). The spike-raster of the SFC-neurons is shown to the right (compare with the upper frame in Fig. 3). These spikes represent the postsynaptic part of the Hebbian learning rule. Generation of the (presynaptic) input time-signal $a(t)$ requires further explanation: in principle arbitrary, sufficiently smooth signals can be used as input. Here, those are extracted from simple line-drawings as indicated in Fig. 6a/b. A spot (rectangle in figure 6a) starts at time zero at some corner of the object and travels along it with constant speed $v$. The velocity vector $\vec{v}(t) = d\vec{r}/dt = (dx/dt, dy/dt)^T$ along the curve $\vec{r}(t)$ is taken as the two-dimensional input function $a(t)$. Hence, every neuron in the SFC has two external inputs, which represent velocity in $x$ respectively $y$-direction. When, during learning, neuron $i$ fires a time $t_i$, its synapses $C_{i1}$ and $C_{i2}$ are set to $dx(t_i)/dt$ respectively $dy(t_i)/dt$. The matrix $C = (C_{ij})$ resulting from such a learning process is shown to the right in Fig. 6; obviously it represents the derivatives of the curve $\vec{r}(t)$ in Fig. 6a (white codes for movement in positive and black for movement in negative $x$ or $y$ direction).[2]

Pattern regeneration is shown in Fig. 6d. Again the SFC has to operate in the stable regime, but now without external input. Instead, the formerly learned synapses are now interpreted as 'output'-synapses controlling the movement in the output space. If a neuron in the SFC fires, its synapses determine the instantaneous velocity vector with which a movement is performed. In Fig. 6d the SFC is activated twice - only the second spike-raster is shown. By choosing different absolute starting positions and different (arbitrary) gain factors the previously stored object

[2] We should mention that the model is not intended as a concrete example for visuo-motor coordination or related tasks, although similarities might exist [32]. Intended is a purely abstract view, just as the standard associative memory at first is an abstract paradigm. Both models may be suited as building blocks for more concrete and complex networks, incorporating static as well as temporal properties of stored entities.

is recovered in two sizes in the input space.

Finally, figures 6e and f display examples for pattern recognition. To this end the thresholds in the SFC-network must be high enough to avoid the stable propagation of excitation without a further external input. An additional temporal input into the actually firing neurons then can lead to the complete recovery of the stored sequence, provided the input pattern matches the synaptic pattern of the actually firing neurons sufficiently closely. At any step only the conjunction of the additional input and that from the previously firing SFC-neuron should lead to the firing of the next cell. If at any position in the SFC, the stored and externally applied snapshots do not mach, the synfire chain in the recognition network dies out. Only when the synfire wave reaches the last node of the sequence (double circle in Fig. 5) the spatio-temporal pattern is recognized. An example for proper recognition of a distorted version of the stored pattern is shown in Fig. 6e. Note that 1) the different parameter settings lead to a faster SFC sequence than before (cf. [55, 60] for speed control of SFCs), 2) the distorsions in the test pattern lead to slight fluctuations in the instantaneous speed of synfire propagation, and 3) the stored pattern effectively uses only 27 of the total of 30 cells in the SFC (cf. matrix C in Fig. 6). Hence, under recognition conditions, the last few nodes do not fire in figure 6e. Finally, Fig. 6f shows failure of recognition for a completely different input pattern.

The above interpretation of synfire activity explicitly takes into account temporal information stored in the network structure. We should note, that it is not very likely that synfire networks of this kind provide a reasonable substrate for *arbitrary* time-patterns an individual might learn, say for example, complex movement patterns. The main reason for this is that sequences longer than some hundred milliseconds need exceedingly large hardware resources [7] and, furthermore, the resulting synfire structure is very inflexible. Motor control (for movements or speech etc.) certainly needs more flexible, probably modular and hierarchically organized structures (cf. also [61, 64]). However, we believe, that chains with roughly some 100 nodes and perhaps 100 neurons per node may be useful storage devices for elementary 'spatio-temporal features' in such architectures: the complete information can still be retrieved in a short time and a time-span of 200 to 500ms would indeed make sense, since this is roughly the duration of syllables or morphemes, which organize speech-production; similar timing intervals have also been proposed to organize other cognitive tasks (see [18] for a collection of related articles).

# 6 From Synfire Chains to Automata

## 6.1 Construction of the deterministic automaton

In light of the definitions of finite automata and regular languages in section 4 it is quite obvious, that the simple spatio-temporal pattern store described above can be interpreted as a realization of a finite automaton. If one neglects that it operates on real valued input (of course input symbols in form of binary spike patterns are also

possible) and that some kind of synchronization mechanism between input symbols and node-firing is necessary (see below), then the scheme in Fig. 5 represents the implemented automaton in analogy to Fig. 4.

In fact, this analogy also shows, that the model is actually capable of recognizing only a single word, the stored temporal pattern. Hence, even if a set of different chains is stored the recognized language always consists of only finitely many words. Grammatical rules, represented by the transition function of the automaton (or alternatively the production rules of the corresponding regular language) are seriously restricted to the form $\delta(z_1, a) = z_2$, where for each $z_1 \in Z$ at most one $a \in \Sigma$ leads to a meaningful target state and all other possible $a$'s to failure. (In the neural implementation the activity simply dies out in case of failure; this might be represented by an extra state in the abstract automaton model.) More complicated syntactical dependencies between symbols are not representable by the spatio-temporal pattern store, which already extends the conventional synfire chain model with respect to the external input. Insofar the computational capabilities of the model are only very poor.

If we want to extend the synfire chain model (respectively the spatio-temporal pattern store) such that at least the functionality of finite state automata is obtained, then a short reflection shows that everything that is essentially missing is the possibility for different target states that can be reached from an actually firing state in dependence of the current input signal. The introduction of this feature leads from a simple linear pattern recognizer to directed and labelled "synfire graph" structures.

The most simple idea for an implementation of a given automaton consists perhaps in a one-to-one translation of the graph representing the automaton into a network of spiking neurons with the same connectivity graph: this requires distinct (synfire-)nodes for every state $z_1, \cdots, z_n$ and every input symbol $a_1, \cdots, a_m$, and direct feedforward connections from the nodes $z_i$ and $a_j$ to $z_k$ for every transition rule $\delta(z_i, a_j) = z_k$. If, for example, all connections have strength one and the thresholds are 1.5 synchronous activation of $z_i$ and $a_j$ then leads to the selective firing of the target node $z_k$. However, this simplest implementation faces the following problem: for two rules $\delta(z_1, a_1) = z_3$ and $\delta(z_2, a_2) = z_3,\ a_1 \neq a_2,\ z_1 \neq z_2$ ambiguities occur such that also the pairs $(z_1, a_2)$ and $(z_2, a_1)$ activate the target node $z_3$. This in general will be an error. (The described problem is well known as the XOR-problem in neural network theory [22].) Hence, the straightforward one-to-one translation of the automaton into a neural network does not work.

To resolve this difficulty we introduce further nodes into the network beside the already mentioned state- and inputnodes. For every transition rule one corresponding *transition node* is needed, giving a total of $m \cdot n$ additional nodes if $n$ is the number of states and $m$ the number of input symbols.[3] Figure 7 shows the principal strategy: on the left a usual projection from node $z_n$ to $z_{n+1}$ is displayed (*synfire-*

[3]It is certainly possible to construct networks with fewer nodes. The solution proposed here, however, is simple and advantageous with respect to the following mathematical analysis.

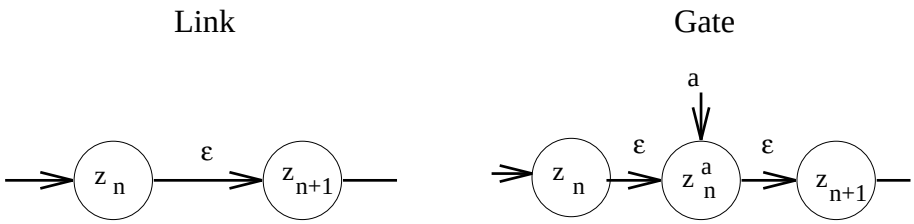


Figure 7: Extension of synfire links to input-dependent transitions.

*link*, compare Fig. 2). The nodes simultaneously represent neurons and states of an automaton. Transitions in the standard model occur spontaneously without further external input; this is indicated by the $\epsilon$ labelling the projection ($\epsilon$ denotes the *empty word* in automata theory). On the right a further node between $z_n$ and $z_{n+1}$ is inserted, called $z_n^a$, which represents the transition rule $\delta(z_n, a) = z_{n+1}$. This transition-node is exclusively acivated by $z_n$ and $a$ such that the above mentioned ambiguities for target nodes with several possible predecessors, cannot longer occur. The transition nodes uniquely represent the entries in the transition function $\delta$ of the automaton. Conceptually every *edge* of the transition graph - for example in Fig. 4 - is replaced by a node that receives input from the respective predecessor and the input node which labels the particular edge. The target of this transition node is the node representing the target state of the respective transition rule.

Figure 8 shows how a deterministic finite automaton can be constructed with the aid of such transition nodes. The intended automaton is the same than that in Fig. 4. The automaton has $n = 3$ states $z_1, z_2, z_3$ and $m = 2$ input symbols 0 and 1 which are represented by nodes $a_0$ und $a_1$. Thus we need $mn = 6$ additional nodes for the possible transitions. These are denoted by 10,11,20,21,30,31 and represent all possible combinations of states and input symbols. Therefore, from every state node $m$ connections of strength $\alpha$ to the respective transition nodes exist, as well as $n$ connections of strength $\beta$ from every input node to their corresponding transition nodes. Furthermore, the 6 transition rules constituting the $\delta$-function of the automaton can be identified as connections $w_{ijk}$ of strength $\gamma$ from exactly one transition node back to one state node. The squares in Fig. 8 at this point of the discussion can be envisaged as simple adders.

We assume that all nodes in the network consist of a single refractory neuron with threshold $\vartheta_1$ for states $z_1$ to $z_n$ and $\vartheta_2$ for transition nodes. In particular this implies that the network operates in continuous time. Synaptic response functions $g(t)$ are assumed to be the same for all connections. Their form as well as the refractory mechanism of the cells to a large degree determines the stable operation of the automaton (section 7).

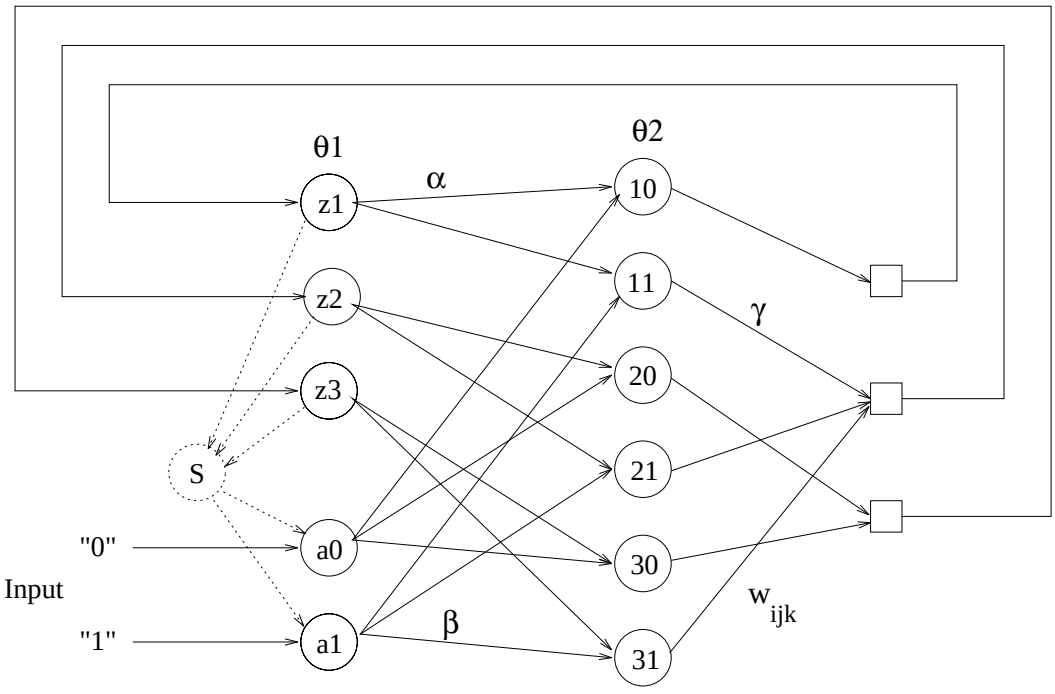


Figure 8: Network of spiking neurons that implements the automaton in Fig. 4. For explanations see text.

### 6.1.1 Proper function of the DFA

It is clear that an arbitrary deterministic finite automaton can be translated into a network of the above described type. In the following we consider function and temporal behavior of such networks in more detail. For simplicity we first assume that $\alpha = \beta = \gamma = 1$, $\vartheta_1 = .75$, $\vartheta_2 = 1.5$, and that the response function g(t) is given by the simple rectangular function $g(t) = \chi_{[\Delta,2\Delta]}(t)$, which is one on the interval $[\Delta, 2\Delta]$ and zero else. Effects due to refractoriness are neglected. Input into the network is delivered by firing of the appropriate input nodes in regular intervals $T = 2\Delta$. Then it is easy to see that the network simulates exactly the intended automaton:

We start at time $t = 0$ by activating the initial state $S$ and the first input node, that is both nodes fire. After the synaptic transmission time $\Delta$ the spikes excite some of the transition nodes. By construction only one of these nodes receives input from both at $t = 0$ firing cells. Due to the particular choice of connection strengths and thresholds, and because the rectangular PSP-function discontinuously jumps to its maximum value at $t = \Delta$, this node fires at time $\Delta$ and no other transition node reaches threshold. Hence, exactly that neuron fires which represents the transition rule applicable in the first step. Because $\gamma = 1 > .75 = \vartheta_1$ after a further delay of $\Delta$ this firing leads to the activation of the target state $z$ which is determined by the

rule $\delta(S, a_1) = z$. This node fires a time $t = 2\Delta = T$ synchronously with the next input symbol. By inductive application of these arguments it follows that in cycle $i = 0, 1, \cdots$ at time $t_i = iT$ always the input node $a_{i+1}$ of an input word $a_1 a_2 \cdots$ fires in synchrony with the actual state node, and at time $t_i + \Delta$ exactly that transition node fires which induces the firing of the correct next state node.

After reading the last input symbol, the input nodes stop firing, but one state node can still fire driven by the last input. If this node represents an accepting state, the word is by definition recognized, otherwise it is rejected. It is easily possible to detect this "halting condition" by a further neuron with threshold .5 which receives synapses of strength 1 from all accepting state nodes and -1 from all input nodes. Firing of this neuron, at a time $\Delta$ after firing of the last state node, would signal that a correct word has been read in. Afterwards no further spiking is possible in the whole network, because without excitation from input nodes no transition node can reach threshold.

This argumentation shows that the network can process arbitrary long input words correctly, that is, it simulates the intended automaton. However, what also becomes clear is, that the usage of purely rectangular PSP-functions reduces the network dynamics essentially to the time-discrete case. Comparable time-discrete models are, for example, discussed in [38, 41].

### 6.1.2 Realistic response functions

The introduction of realistic response functions leads to several problems. Such PSPs are characterized by a synaptic transmission time, a smooth, not infinitely fast, rising edge and an exponential decay (see (1)). Problems are particularly caused by the exponential decay, since after one run of activity through the network the potentials will in generally have not been relaxed to zero, at least not for plausible choices of PSP time constants. Thus temporal summation occurs on membranes, which in the worst case leads to firing of erroneous cells, but in any case shifts firing times on the rising edge of the response functions. It is no longer obvious that the proper function of the network can be guaranteed for arbitrary input sequences.

Typical synaptic delays are roughly one and rise times up to a few milliseconds; decay constants can be several times larger than rise times. Because only two synaptic transmissions are involved and the transmission is fast, the cycle time of the automaton, i.e. the time between firing of one state node and the next, is of the order of the decay time or smaller. Therefore considerable temporal summation on membranes must be expected.

To repair this problem we have to introduce further delays which effectively increase the cycle time. One possibility for this is to insert futher nodes at the sites indicated by squares in Fig. 8. Each square is replaced by a chain of $k > 0$ refractory neurons coupled feedforward with strength $\gamma$ and threshold $\vartheta_1$. Then a lower bound for the cycle time is $T_m \geq (k + 2)\Delta$ where $\Delta$ is the synaptic delay. Since any particular neuron fires at most once in a given cycle it is clear that by choosing $k$ large

enough the error potentials of *all* cells after one run of activity trough the network can be made uniformly smaller than an arbitrary small given bound. Moreover, for appropriate thresholds the delays per stage can be considerably larger than $\Delta$ (as assumed in the lower bound for $T_m$). If $\tau$ denotes the rise time of PSPs, delays up to $\Delta + \tau$ per stage can be reached. Since the decay of PSPs is exponential, about 20 milliseconds should be sufficiently large cycle times corresponding with frequencies in the gamma-range and perhaps 5 or 7 synaptic stages in the network. For cycle times in this range, which simultaneously presents the shortest possible interspike interval, refractoriness of cells should play no significant role and may be neglected in the sequel.

Even if error potentials are made small by an appropriate number of delay stages they nonetheless shift firing times in a hard to predict pseudo-random manner. Moreover cortical neurons always receive considerable excitation due to stochastic background activity which adds to the membrane fluctuations. Therefore, cycle times in the network will not be constant as in the 'time discrete' model with rectangular PSPs. Instead they will scatter in a certain range. This requires some kind of synchronizing mechanism between the firing of input nodes and the network dynamics to ensure that also for long input words the nodes representing the kth input symbol and state fire synchronously within certain bounds.

The necessary synchronization can be reached by several means. One way is to derive some kind of 'read' signal directly from the network activation, for example, the firing of the state nodes. This signal can excite the input neurons in form of a global threshold control enforcing firing of the next input node. Which input state fires will be further determined by another (here not explicitely specified) neuronal network sending subthreshold activation to the input nodes. This possibility is indicated by the dotted 'synchonization path' in Fig. 8. If delay nodes are required, as it will usually be the case, then the synchronizing signal should be derived as late as possible from cells firing in the delay lines, because spike shifts in earlier stages can be completely ignored. This kind of input synchronization might be supported by bidirectional connections between cortical sites as they are known to exist within and between many cortical areas.[4]

Instead of deriving trigger signals for the next input symbol from the network itself, one may alternatively imagine that an independent input rhythm provides the firing of input nodes driven by oscillations in some external network. Interestingly it is possible that the intrinsic dynamics of the automaton synchronizes with the rhythmic input supposed the internal and external cycle times are not too different (cf. section 7). This does not even require that the input intervals are strictly periodic; some scatter in spike times is well tolerable and only an approximate

[4]Regarding the primary visual area V1 and the lateral geniculate nucleus (LGN) in thalamus, which provides the main ascending input to V1, it is known that synchronous firing of cells in V1 is reflected by an accompanying synchronization of LGN activity due to mutual connections. This way V1 organizes its own input into synchronized volleys of spikes in the gamma-range ([49], cf. also [54].)

synchrony is required. According to the above estimate for the necessary cycle times of the automata suitable inputs could be supplied by network oscillations in the gamma-range.

It section 7 we prove formally, that both versions of input synchronization enable the processing of arbitrary long input sequences. Thus synfire graph networks can simulate the intended automata also when realistic response functions are used, provided sufficiently many delay stages are added. The necessary number of such stages is independent of the particular automaton and a certain amount of bounded membrane noise is admissible as well as a bounded scatter of input spike times. Moreover instead of single cells, the nodes of the network may also consist of pools of cells. In this case all cells within a pool fire in a certain time-interval which can be kept bounded by a constant for arbitrary long input sequences. A proof of these propositions is given in section 7.1.

### 6.1.3 Nondeterministic automaton

As already mentioned earlier nondeterministic finite state automata can also be simulated by networks similar to that in Fig. 8. We have seen that in case of the deterministic automaton in every cycle exactly one state and one transition node as well as the corresponding (unique) chain of delay nodes become activated. This is equivalent to a single propagating wave of activity in the synfire chain (respectively now 'synfire graph') picture. The path through the network taken by the wave is controlled by the external input. A possible nondeterminism is expressed by the fact that several states can be active at any given time. This corresponds with several propagating waves. Accordingly at time zero all initial states in the set S (see section 4) have to be activated simultaneously with the first input node. Whether an input word is accepted can be detected similar than for the DFA by means of a special neuron receiving connections of strength one from all accepting states and those of strength $-n$ from all input nodes, the latter because during reading of the input symbols only a single input node fires per cycle, but up to $n$ state neurons. The threshold of this neuron can again be $.5\hat{g}$.

Furthermore the connections $w_{ijk}$ in Fig. 8 have to be chosen differently for NFAs in comparison with DFAs. This is, because a state can now have a varying number of possible target states (between 0 and $n$) for one and the same input symbol. For every alternative $\delta(z_i, a_j) = z_k$ the respective connection $w_{ijk}$ must be set to $\gamma$ (otherwise to zero). Thus, an arbitrary transition node now has between zero and $n$ outgoing connections in contrast to the previous case, where these cells contacted exactly one target. These connections again correspond with entries in the transition function of the automaton.

From the above it follows that the number of active neurons respectively propagating waves depends on time. Excited waves melt together if two transition rules actually lead to the same target node. Contrary, new waves are evoked if several successor states branch from an active state for the current input. This is an expression

of the 'nondeterminism' of the computation.

As for the DFA it is not difficult to see that the network with simple rectangular response functions $g(t) = \chi_{[\Delta,2\Delta]}$ and input times $t_i = 2\Delta i$ for the symbol $a_{i+1},\ i = 0, \cdots, l-1$ simulates exactly the intended NFA. We skip the details at this point. Furthermore, using realistic response functions it is again possible to prove that the versions of input synchronization distinguished for the DFA also enable stable processing of arbitrary input sequences for the NFA-network. Again pools, additional membrane fluctuations and scatter in the timing of input spikes are allowed. This is proved in section 7.2. The stability and synchronization problems are similiar than before and can be solved by inserting an appropriate number of delay stages. In contrast to the DFA the necessary number of such stages now depends on the automaton under consideration, particularly on the number of its states. The reason for this is that a varying number of spikes of transition nodes can converge on a given target node. Therefore the compound PSPs can be up to $n$ times larger than before. Accordingly the potentials need a longer time to relax below a given fixed bound. The number of necessary delay stages is of the order $\log(n)$. This problem could be avoided by introducing a nonlinear saturation of the compound PSPs. Obviously it does not occur if PSP functions vanishing for times larger than some fixed time $t_m$ are used as in [33, 34].

An additional problem occuring in the NFA network but not the DFA is the following. If several waves of activation are propagating, different target neurons of transition nodes can receive a varying number of input spikes in the same operation cycle. Therefore these neurons will fire at different times. The additional scatter in spike times due to this unpredictable effect can become so large that a proper function of the automaton cannot longer be garanteed. However, it is possible to make the additional scatter small, if, first, the membrane fluctuations are made small by an appropriate number of delay stages, and, second, the firing thresholds of the target neurons are chosen only slightly above the maximally expected fluctuations. This way it can be reached that the target neurons fire within an arbitrary short fixed interval after the synaptic transmission delay is over independently of the number of activating spikes. This again requires $k = O(\log(n))$ delay stages. Furthermore, if additional noise sources are taken into consideration then with increasing $k$ the condition on the bound of this noise becomes increasingly restrictive, because the scatter in spike times accumulates with an increasing number of stages. This means, theoretically it is possible to ensure a stable operation for certain parameter regimes, but in physical realizations of the network architecture large automata are perhaps not easily implementable, because the external noise sources in general cannot be controlled, that is, made as small as required.

In any case response functions with a steep rising edge would be of advantage, since the scatter in spike times is roughly inversely proportional to the rise time of the PSPs (and proportional to the membrane fluctuations). With that respect decaying exponentials (or first order low-pass membranes) and rectangular PSPs provide rather special cases. Inasmuch as also time-discrete networks in some sense

utilize PSPs with infinite slope the problem does not occur in these models (it only does insofar the transfer of results found for time-discrete automata to more realistic models is made uncertain).

# 7 Stability proofs

In this rather technical section we show that the above described automata models can indeed process input sequences of arbitrary length even if biologically plausible response functions are used. The principal correctness of the operation has been already shown for rectangular PSPs. Hence, we can restrict the discussion in the following to dynamical aspects of stability and synchronization.

We require that the response functions $g(t)$ are (i) continuous (ii) equal to zero for $t \leq \Delta < \infty$ (iii) continuously differentiable and strictly monotonously increasing on $[\Delta, \Delta + \tau]$ for some constant $\tau$ with $0 < \tau < \infty$ and (iv) exponentially bounded, that is there exist constants $C_1 > 0,\ C_2 > 0$ such that $g(t) < C_1 \exp(-C_2 t)$ for all $t \in \mathbb{R}$. Furthermore let (v) $\hat{g} := \max_{t \in \mathbb{R}} g(t) = g(t + \tau)$ and $g_+^{-1}(x)$ be the inverse of the strictly monotonously increasing branch according to (iii). The constants $\Delta$ and $\tau$ are the synaptic transmission time respectively the rise time of the PSP. $1/C_2$ provides a lower bound for the decay time constant. The alpha function (1) can be envisaged as a prototypical response function.

## 7.1 Deterministic Automaton

In section 6.1 we stated that after one cycle through the network the membrane potentials in general will not have been relaxed to zero. This can lead to the firing of wrong nodes as well as shifts in spike times. In the worst case a particular synapse is activated in every cycle; then the temporal summation is maximal. Let $0 = t_0 > t_1 > t_2 > \cdots$ be the firing times of the exciting neuron. A cycle needs a certain minimal time, which since two synaptic transitions are involved must be larger or equal than $T_m = 2\Delta$, thus $|t_i - t_{i+1}| \geq 2\Delta$. The same is assumed to hold for the input times too (see below). For $t > 0$ the response to the last spike at $t_0 = 0$ is the signal and responses to earlier spikes constitute an error potential not yet decayed to zero. Now, PSP functions are exponentially bounded. Thus for $t > 0$ we have

$$\sum_{i=1}^{\infty} g(t - t_i) \leq C_1 \sum_{i=1}^{\infty} \underbrace{\exp\left(-C_2 T_m \cdot i\right)}_{=:q^i\ ,\ 0<q<1} = C_1 \sum_{i=1}^{\infty} q^i = C_1 \frac{q}{1-q} =: \hat{b}\ . \tag{5}$$

This provides an upper bound for the maximally possible contribution of the fluctuations to the membrane potential that a single synapse of strength one can evoke on a target cell for $t > 0$. The definition $q = \exp(-C_2 T_m)$ shows that $q$ and hence also $\hat{b}$ decrease rapidly when the cycle time increases. As explained in section 6.1 the cycle time can be increased by additional delay stages. If $k$ such stages are

present the cycle time is at least $T_m = T_m(k) = (k+2)\Delta$. Since this is also a lower bound for the shortest possible interspike interval we can furthermore neglect refractory effects if $k$ is large enough. In the sequel we therefore assume that the influence of refractoriness is vanishing small. (More formally we could also impose an exponential bound on the refractory mechanism, i.e. the dynamical threshold $\vartheta(t)$, and integrate the effects in $\hat{b}$).

Even if the error potentials are small they shift spike times of cells. We have to show that those remain in certain bounds for arbitrary long input sequences.

Assume the error potentials are exactly zero and that at time $t = 0$ a state node and an input node fire synchronously. Then the corresponding transition node fires at time $t_0 = g_+^{-1}(\vartheta_2/(\alpha+\beta))$ (provided the thresholds are chosen appropriately). Firing of this node evokes a spike in the first delay node after a time $t_1 = g_+^{-1}(\vartheta_1/\gamma)$; further delay nodes as well as eventually the next state node also fire at intervals of $t_1 = g_+^{-1}(\vartheta_1/\gamma)$. Hence, one cycle through the network ideally needs the time $T = t_0 + (k+1)t_1$.

To avoid unnecessary notational complications in the sequel we set $\alpha = \beta = \gamma = 1, \vartheta_1 = .75\hat{g}$ and $\vartheta_2 = 1.5\hat{g}$. Then $t_0 = t_1$ and because $g'(t_0) > 0$ and $g'$ is continuous on $]\Delta, \Delta+\tau[$ there exists a neighborhood $U = ]t_0 - \delta, t_0 + \delta[ \ \subset \ ]\Delta, \Delta+\tau[$ where $g'$ is strictly positive. We define $g_1 = \min_{t\in U} g'(t)$ and $g_2 = \max_{t\in U} g'(t)$ and require that $\delta > 0$ is so small that $g_2/g_1 < c_4$ for some constant $c_4 > 1$. This choice of $\delta$ is possible for any real $c_4 > 1$ because the continuity of $g'(t)$ in a neighborhood of $t_0$ implies that the limits $\lim_{\delta\to 0} g_1$ and $\lim_{\delta\to 0} g_2$ are both equal to $g'(t_0) > 0$, hence $\lim_{\delta\to 0} g_2/g_1 = 1$.

The influence of PSPs of earlier spikes can be made uniformly small, i.e. for all synapses simultaneously, by sufficiently many delay stages. We take $k$ so large that the error potential $b_s(t)$ evoked by a single synapse of strength one on its target for times larger than $T_m(k) = (k+2)\Delta$ after the last firing is absolutely bounded by a constant $b > 0$ with $b < \hat{g}/8$ and

$$(k+2)b/g_1 < c_3\delta \tag{6}$$

for some constant $c_3 > 0$. Since according to equation (5) the error potentials decay exponentially in the number of delay stages such constants $k$ and $b$ always exist. Since no neuron receives more than two spikes in any cycle and the convergence is uniform, $k$ and $b$ can in particular be chosen independently of the automaton under consideration. The first condition on $b$, $b < \hat{g}/8$, in the following ensures that the correct transition nodes still fires (and no other) even under noisy conditions, and the second, eqn. (6), implies that the scatter in spike times remains in bounds.

Now we assume that one state and one input node do not fire exactly at time zero but deviate by $dt_1$ respectively $dt_2$ from zero. We require that $|dt_1| < c_1\delta$ and $|dt_2| < c_2\delta$ for constants $c_1 > 0, c_2 > 0$ with $c_1 + c_2 + c_3 < 1$.

We show that after a time near the ideal cycle time $T = (k+2)t_0$ the next state neuron reliably fires and the deviation of its firing time from $T$ remains within certain error bounds. To this end define $G(t, dt_1, dt_2) := g(t - dt_1) + g(t - dt_2) + b_1(t) + b_2(t)$

which is the potential of the transition neuron addressed by the firing of the state and input node. $b_1(t)$ and $b_2(t)$ represent contributions of the two synapses to the membrane fluctuations (note that $\alpha = \beta = 1$). Because $\max(\alpha, \beta)\hat{g}+2b < 5/4\hat{g} < \vartheta_2$ at most this transition neuron can fire and no other. With $t = t_0+dt$ Taylorexpansion of the PSP-functions in G gives for arbitrary $dt, dt_1, dt_2$ with $|dt - dt_i| < \delta,\ i = 1, 2$:

$$G(t_0+dt, dt_1, dt_2) = \vartheta_2+g'(\tilde{t}_1)(dt-dt_1)+g'(\tilde{t}_2)(dt-dt_2)+b_1(t_0+dt)+b_2(t_0+dt)\ . \quad (7)$$

Here we have used $2g(t_0) = \vartheta_2$. The $\tilde{t}_i,\ i = 1, 2$ are appropriate times (depending on $dt - dt_i$) in the intervals $\tilde{t}_i \in [t_0, t_0 + dt - dt_i],\ i = 1, 2$.

Assume for the first $dt_1 \geq dt_2$ and $dt = dt_1 + b/g_1$. Hence $|dt - dt_1| < c_3\delta < \delta$ and $|dt - dt_2| = |dt_1 - dt_2 + b/g_1| < (c_1 + c_2 + c_3)\delta < \delta$, and from equation (7) we obtain the estimate

$$\begin{aligned} G(t_0 + dt_1 + b/g_1, dt_1, dt_2) &= \vartheta_2 + g'(\tilde{t}_1)b/g_1 + g'(\tilde{t}_2)(dt_1 - dt_2) + g'(\tilde{t}_2)b/g_1 + \quad (8)\\ &+b_1(t_0 + dt) + b_2(t_0 + dt)\ >\ \vartheta_2 + g_1(dt_1 - dt_2) + 2b - 2b\ \geq\ \vartheta_2\ . \end{aligned}$$

On the other hand for $dt = dt_2 - b/g_1$ the same argument implies $G(t_0 + dt_2 - b/g_1, dt_1, dt_2) < \vartheta_2$ and from both estimates together it follows that there exists a $t_0'$ in the interval $]t_0+dt_2-b/g_1, t_0+dt_1+b/g_1[ \subset U$ with $G(t_0', dt_1, dt_2) = \vartheta_2$. The same holds when we assume $dt_1 < dt_2$. Therefore under the above conditions it is ensured that the addressed transition neuron still reaches threshold. The threshold crossing need not be uniquely defined, particularly not, if further noise sources are taken into consideration (see below). But the above arguments imply that all threshold crossings are contained in the neighborhood $U$. Only the first leads to the firing of the cell, later firings are prohibited by the refractory mechanism. Let $t_0'$ be the firing time. Then (7) with $t_0' = t_0 + dt$ and $G(t_0', dt_1, dt_2) = \vartheta_2$ imply the bound

$$|dt| = \left| \frac{g'(\tilde{t}_1)dt_1 + g'(\tilde{t}_2)dt_2 - b_1(t_0') - b_2(t_0')}{g'(\tilde{t}_1) + g'(\tilde{t}_2)} \right| \leq \frac{g_2}{2g_1}(|dt_1| + |dt_2|) + \frac{b}{g_1} \quad (9)$$

for the shift of the firing time $t_0'$ in comparison with the ideal value $t_0$.

Similarly one finds that fluctuations in the following stages lead to additional shifts which (per stage) are smaller than $b/g_1$. Thus, given a total of $k + 2$ stages, the next state node will fire at a time $t = T + dt_1'$ where

$$|dt_1'| < \frac{g_2}{2g_1}(|dt_1| + |dt_2|) + (k + 2)\frac{b}{g_1} < \left(\frac{c_4}{2}(c_1 + c_2) + c_3\right) \cdot \delta\ . \quad (10)$$

For the time $dt_1$ of the state firing near zero we assumed $|dt_1| < c_1\delta$. If we require at least the same accuracy for the firing of the state after one cycle equation (10) leads to the inequality $c_4(c_1 + c_2)/2 + c_3 \leq c_1$ as a condition that deviations in spike times of state nodes do not increase in one cycle. The constants $c_1$ to $c_4$ must further satisfy the above stated conditions $c_1 + c_2 + c_3 < 1,\ c_i > 0, i = 1, 2, 3$ and $c_4 > 1$. One possible choice is $c_1 = 1/2,\ c_2 = 1/6,\ c_3 = 1/12$ and $c_4 = 5/4$.

Now, let $a_1 a_1 a_2 \cdots a_l$ be an input word of length $l$. We consider different alternatives for the synchronization between input symbols and the network dynamics.

1) 'Quasiperiodic input': Here the initial state is activated at time $t \in ]-c_1\delta, c_1\delta[$ and the input node $a_i$ at an arbitrary time in the interval $](i-1)T - c_2\delta, (i-1)T + c_2\delta[,\ i = 1, \cdots l$ with $T = (k+2)t_0$ as before. The input is assumed to be supplied by some external influences. Furthermore it is rhythmic with period T, but some scatter in the ideal input times $(i-1)T$ in the range $\pm c_2\delta$ is admissible. Inductive application of the above argumentation then shows that the network operates stably on arbitrary long input sequences, that is, the scatter of firing times of state nodes around the ideal times $iT,\ i = 0, 1, 2, \cdots, l$ remains absolutely bounded by $c_1\delta$.

2) 'Statistical input intervals': For the input times it is required that the time between firing of successive input nodes is given by $T_i = T + dt_2(i)\ i = 1, \cdots, l-1$ where the deviations $dt_2(i) \in ]-c_2\delta, c_2\delta[$ are arbitrary. In contrast to the previous case this includes the situation where the input cycle is systematically somewhat shorter or longer than T. Moreover the input times in 1) are strongly correlated over arbitrary long times, whereas correlations can decay in the present case, because independent and randomly fluctuating deviations accumulate roughly proportional to $\sqrt{i}$.

Assume that at time $dt_2$ the input node $a_1$ fires and at time $dt_1$ the initial state. As before let $|dt_1| < c_1\delta$ and $|dt_2| < c_2\delta$. Now w.l.o.g. we can in this and every following cycle redefine the time origin to coincide with the spike time of the actually firing input node. With respect to the new time origin the transformed spike deviations are $\widetilde{dt}_2 = 0$ and $|\widetilde{dt}_1| \leq |dt_1| + |dt_2| < (c_1 + c_2)\delta$. The next input by assumption occurs at time $T + dt_2(1)$ relative to the new origin and the next state node fires according to the above argumentation at a time $T + dt_1(1)$ in the interval $T \pm dt_1'$ with $dt_1'$ given by (10) but now using the transformed differences. Equation (10) again leads to $dt_1' < c_1\delta$ with $c_1$ to $c_4$ as before. Redefining the time origin again to the firing time of the input neuron $T + dt_2(1)$ we find $\widetilde{dt}_2(1) = 0$ and $|\widetilde{dt}_1(1)| \leq |dt_1(1)| + |dt_2(1)| < (c_1 + c_2)\delta$. Inductively it follows that the scatter in spike times of the state neurons relative to the input neurons is now bounded by $(c_1 + c_2)\delta_1$. Thus the internal dynamics of the network synchronizes with the input process and the automaton can again process arbitrary long input sequences.

3) 'Input request by the network': This possibility is indicated by the dashed 'synchronization path' in Fig. 8. Here, trigger signals are derived from the network activity itself (particularly the state nodes in Fig. 8), which enforce firing of a new input node. Proper function of this mechanism requires that the cycle time T is about the same than the time a signal needs to propagate through the synchronization branch and evoke the next input (plus/minus error bounds as before). Since we do not specify the dynamics of the input neurons in detail satisfaction of the error bound is a requirement imposed on these input circuits. Clearly, the trigger signals need not be derived from the state nodes. If delay nodes are present it is more advantageous to derive them from nodes as late as possible in the delay line, because all previous spike shifts can be neglected.

## 7.2 Nondeterministic Automaton

Stability and synchronization problems for the NFA are similar to those discussed for the DFA. In the following we only sketch some points where differences appear.

If approximately at time $t = 0$ a total of $n_0$ state nodes fire all within the interval $[-dt_1, dt_1]$ and an input node fires with precision $dt_2$, then exactly $n_0$ transition nodes become activated in the interval $[t_0 - dt, t_0 + dt]$. The times $t_0$ and $dt$ are the same than for the DFA because all activated nodes receive exactly one spike from an input node and one from a state node.

Differences appear when we consider the projection from transition nodes to their targets. These cells now can experience a varying number of up to $n$ incoming spikes. They must fire when at least one input spike is present because the input spike represents the fact that the corresponding transition rule can (and hence must) be applied in the present situation. This requires $n\hat{b}\gamma < \vartheta_1 < \gamma\hat{g}$.

It is clear that error potentials can be made small as before by inserting delay stages, but because the compound PSPs can now be at worst $n$ times as large as for the DFA the number of necessary delay stages will depend on $n$, the number of states in the automaton. The worst case occurs when $n$ spikes converge repeatedly on the same neuron. If we require that the error potentials are bounded by a constant $B$ we have at worst $n\hat{b} < B$ with $\hat{b}$ from (5) as an upper bound for the contribution of a single synapse of strength one. $q$ decays exponentially in $k$; thus $1 - q$ asymptotically approaches 1. Then (5) implies an asymptotically logarithmic dependence of the necessary number of delay stages from $n$, i.e. $k \sim 1/(C_2\Delta)\ln(nC_1/B)$. This means, we can bound the error potentials, but the neccessary number of stages depends on the size of the automaton. We should further mention that $B$ cannot be chosen arbitrarily. As outlined below, it has to satisfy constraints analogous to (6). Obviously the discussed problem does not occur when response functions are used that are different from zero only on a finite interval, for example rectangular PSPs or other functions with finite support [33]. Then a constant number of delay stages is enough, if the total delay exceeds the duration of the PSP.

If different target neurons of transition nodes simultaneously experience a different number of incoming spikes a second problem occurs, because this can lead to an increase in the scatter of spike times. The increase can be held small if the firing thresholds are chosen just above the maximally expected error potentials $\gamma B$. This obviously prevents wrong cells from firing. The situation for correctly addressed transition nodes is depicted in Fig. 9 where $b := B/n$. If all spikes of transition nodes fall into the interval $[0, dt]$ then the earliest possible firing time of a target neuron is $t_{min} = \Delta$. This lower bound is approached the more cells fire at $t = 0$ and converge on the particular neuron. Contrary the latest possible firing time is determined by a single spike of one transition node at time dt which converges onto a target neuron with zero error potential. One finds $t_{max} = dt + g_+^{-1}(\vartheta_1/\gamma)$. At relatively high thresholds as in Fig. 9 the possible increase in the scatter of firing times $\widetilde{dt} = t_{max} - t_{min} - dt$, in addition to the scatter $dt$ already present in the previous

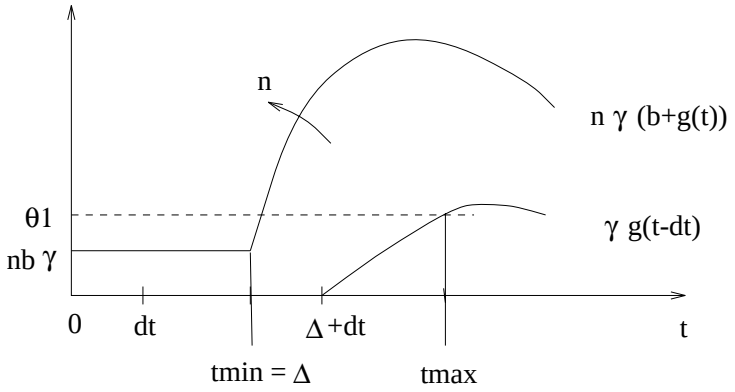


Figure 9: Minimal and maximal firing times at the projection from transition nodes to their targets. A vanishing small scatter in spike times requires that $\vartheta_1$ and therefore also $b$ tend to zero.

stage can be considerably large. However, since $B = nb$ can be made arbitrarily small, $\vartheta_1$ can be chosen arbitrarily close above $nb\gamma$, and $\lim_{s\to 0} g_+^{-1}(s) = \Delta = t_{min}$ holds, one can again reach that the scatter increases by a neglectable amount only. The increase $\widetilde{dt}$ is furthermore the smaller the larger the slope of the PSP functions is. Hence, fast rise times support the proper function of the automaton. In the limiting case of a discontinuity in $t = \Delta$, i.e. rectangular PSPs, the discussed problem vanishes completely [38, 41]. Moreover, observe that it only occurs for the projection from transition nodes to their targets. Only at this point a variable number of incoming spikes can influence the spike scatter. In later stages thresholds can therefore be chosen larger, such that larger transmission delays are obtained. As for the DFA it is then possible to prove that the there distinuished versions for input synchronization also enable the stable processing of arbitrary long input words in the NFA network. To this end we finally note that $b$ in analogy to (6) must satisfy the condition $\widetilde{dt} + (k+1)b/g_1 < c_3\delta$. With this choice the proof follows exactly the same lines than for the DFA with the same constants $c_1 \cdots c_4$. This condition restricts the choice of $b$ such that $k$ must perhaps be further increased in comparison with the above estimate. Nonetheless the asymptotic number of necessary stages can again be shown to depend logarithmically on $n$.[5]

[5] If, for example, g(t) in a region just above $t = \Delta$ has everywhere a slope of at least $g_1'$ and the firing time falls into that region, then $\widetilde{dt} \leq n\hat{b}/g_1'$, and using (5) and some simple approximations one finds $k \sim \frac{2}{C_2\Delta} \ln\left[\frac{C_1}{C_2\Delta c_3\delta}\left(\frac{2}{g_1} + \frac{n-1}{g_1'}\right)\right] = \mathcal{O}(log(n))$. (The same expression with $n = 1$ can also be obtained for the DFA.)

## 7.3 Cell pools and noise

As a further step towards more biological realism we consider the case where synfire nodes consist of cell pools instead of just a single neuron. Furthermore the influence of additional background noise is discussed.

First, let us introduce cell pools. Here, every node in the network is replaced by a number of cells, not necessarily the same number for different nodes. The architecture of the network itself is not changed, but if a connection exists between two nodes and the nodes contain $n$ respectively $m$ neurons, then this connection is replaced by the projection (cf.Fig. 2)

$$\frac{1}{m}\sum_{j=1}^{m} C_{ij}(g(t-t_{ij})+b_{ij}(t)),\ i=1\cdots n\ . \tag{11}$$

Note the 'normalization' of the sum by $1/m$, the number of cells in the source node. With this choice thresholds can be chosen constant independently of the poolsizes. We take the same thresholds $\vartheta_1, \vartheta_2$ than before for all neurons in the respective pools. $C_{ij}$ in (11) are synaptic strengths, $t_{ij}$ the last firing times of neurons in the source node, and the $b_{ij}(t)$ represent the error potentials due to earlier firing of source cells. The automata certainly will not work properly for arbitrary choices of the connections $C_{ij}$. Therefore, we first assume that the $C_{ij}$ in (11) are independent and identically distributed random variables and the poolsize tends to infinity. Afterwards a finite poolsize is discussed as well as two important special cases: identical couplings and diluted projections.

**Random connectivities in the asymptotic limit.** We consider a single projection of the form (11) and assume that the $C_{ij}$ are independent and identically distributed random variables with expectation $EC_{11} = C$ and $EC_{11}^6 < \infty$. The latter requirement is exclusively necessary for technical reasons. It is, for example, satisfied whenever the distribution function is continuous and has finite support; in addition finitely many pointmasses at particular synaptic strengths (e.g. at strength zero) are also admissible in the distribution function. Because, physiological values for synaptic efficacies are certainly bounded, all reasonable distribution functions satisfy the stated condition. Without loss of generality we may assume that $C$ is one. Depending on the type of the projection $C$ may be identified with $\alpha$, $\beta$ or $\gamma$ in section 7.1. (Remember that we have used $\alpha = \beta = \gamma = 1$ in section 7.1 for simplicity.) Let $w_{ij} := C_{ij} - C,\ i,j = 1,2,3\cdots$. In the sequel we repeatedly refer to the following lemma.

*Lemma 1. Let $w_{ij},\ i,j = 1,2,3,\cdots$ be an infinite collection of i.i.d. random variables with $Ew_{11} = 0$ and $Ew_{11}^6 < \infty$. Then $\overline{\lim}_{n\to\infty} \sup_{i=1\cdots n} \left|\frac{1}{n}\sum_{j=1}^{n} w_{ij}\right| = 0$ a.s.*

A proof of the lemma can be found in [59]. Note that it is stronger than the usual law of large numbers inasmuch as it ensures almost sure (or strong) convergence for

an infinite collection of sums of random variables and not just a single sum.

Now, (11) can be written as

$$C\frac{1}{m}\sum_{j=1}^{m} g(t-t_{ij}) + \frac{1}{m}\sum_{j=1}^{m} C_{ij}b_{ij}(t) + \frac{1}{m}\sum_{j=1}^{m}(C_{ij}-C)g(t-t_{ij}) \ . \tag{12}$$

The second term can be absolutely bounded by the Holder inequality (observe that the $C_{ij}$ are positive)

$$\left|\frac{1}{m}\sum_{j=1}^{m} C_{ij}b_{ij}(t)\right| \leq \left(\frac{1}{m}\sum_{j=1}^{m} C_{ij}\right)\sup_{ij}|b_{ij}(t)| \ . \tag{13}$$

$\sup_{ij}|b_{ij}(t)|$ can be bounded for times larger than the mimimal cycle time by an arbitrary small $b$, because additional delay stages lower the error potentials uniformly for all synapses. Then the lemma implies in the asymptotic limit $\overline{\lim}_{n\to\infty}\sup_{i=1\cdots n}\left|\frac{1}{n}\sum_{j=1}^{n} C_{ij}b_{ij}(t)\right| \leq Cb$ a.s.

The same argument cannot directly be applied to the third term in (12) to conclude that it must vanish asymptotically. This is, because the firing times $t_{ij}$ in general are not independent; particularly they depend in a complicated manner on the $C_{ij}$. However, if we assume that $t_{ij} = 0$ for all $i, j$, i.e. synchronous firing of all cells in the source node, then $g(t)$ can be moved outside the sum in the third term and the almost sure convergence towards zero holds uniformly in $i$ (that is, for all cells in the target node simultaneously). Furthermore, if all $t_{ij}$ are zero the first term in (12) simply reduces to $Cg(t)$. Putting things together, *supposed* all cells in the source node fire synchronously, we can conclude that in the asymptotic limit a random projecion (11) behaves with arbitrary precision and probability one like a connection between two individual cells in the previously considered automata comprising single neurons per synfire node. This particularly implies, that the ideal transmisson times $t_0$ and $t_1$ defined in section 7.1 for vanishing fluctuations $b_{ij}(t)$ are necessarily the same for random projections (11) in the asymptotic limit. Moreover, since only a finite number of nodes and projections exist in an implementation of an arbitrary finite automaton, the almost sure convergence can be reached uniformly for all projections (the union of finitely many sets of measure zero again has measure zero). Hence, although in the sequel we only consider single projections the reader should bear in mind that convergence proofs immediately carry over to the whole set of projections in any implementation of a finite automaton.

Now assume non-ideal conditions $t_{ij} \neq 0,\ b_{ij} \neq 0$. As for the DFA-network of single neurons we have to show that, first, the cells in the target node still fire (and no others do) and, second, the firing times of all cells in one node remain in certain bounds.

We only consider the situation for transition nodes in detail. Projections in further stages can be handled similarly. Transition nodes receive input through two projections of the form (11), one from an input node the other one from a state

node. For simplicity we assume that all nodes contain the same number of $m$ cells. The potentials of the cells $i = 1, 2, \cdots, m$ in the target node then read

$$G_i(t, \vec{t_1}, \vec{t_2}) = \frac{1}{m} \sum_{j=1}^{m} \left[ C_{ij}^{(1)} \left( g(t - t_{1j}) + b_{ij}^{(1)}(t) \right) + C_{ij}^{(2)} \left( g(t - t_{2j}) + b_{ij}^{(2)}(t) \right) \right] \ , \quad (14)$$

where $\vec{t_l} = (t_{l1}, \cdots, t_{lm})^T,\ l = 1, 2$. The index $l$ refers to the projection from the input respectively state node.

It is not difficult to see that at most one transition node can fire in response to the conjunctive firing of one input and one state node around time zero. This, of course, is the transition node receiving input from both these source nodes. All other transition nodes receive input from at most one of them. Hence for those

$$\left| G_i(t, \vec{t_1}, \vec{t_2}) \right| \leq 2b + \hat{g} \max_{l=1,2} \left( \frac{1}{m} \sum_{j=1}^{m} C_{ij}^{(l)} \right) \quad (15)$$

and the lemma implies that asymptotically

$$\overline{\lim_{n \to \infty}} \sup_{i=1 \cdots n} \left| G_i(t, \vec{t_1}, \vec{t_2}) \right| \leq 2b + \hat{g} \max_{l=1,2} E C_{11}^{(l)} = 2b + \hat{g} \max(\alpha, \beta) < \frac{5}{4} \hat{g} < \vartheta_2 \ . \quad (16)$$

Therefore at most the addressed transition node can fire.

Now we consider the addressed transition node. As in section 7.1 we can expand $G_i(t, \vec{t_1}, \vec{t_2})$ into a Taylorseries around the ideal time $t_0$

$$G_i(t_0 + dt, \vec{t_1}, \vec{t_2}) = \vartheta_2 + \frac{1}{m} \sum_{l=1}^{2} \sum_{j=1}^{m} C_{ij}^{(l)} \left( g'(t_{ij}^{(l)})(dt - t_{lj}) + b_{ij}^{(l)}(t_0 + dt) \right) \ . \quad (17)$$

The expansion is valid if $|dt - t_{lj}| < \delta$ for $l = 1, 2$ and all $j = 1, 2, \cdots, m$. The $t_{ij}^{(l)},\ l = 1, 2;\ i, j = 1, 2, \cdots, m$ are appropriate times in the intervals $[t_0, t_0 + dt - t_{lj}] \subset U$. ($\delta$ and U are the same than defined in section 7.1.)

Requiring $t_{lj} < c_l \delta,\ l = 1, 2$ and choosing $dt_> = \max(c_1, c_2)\delta + b/g_1$ one derives from (17) that $G_i(t_0 + dt_>, \vec{t_1}, \vec{t_2}) > \vartheta_2$ for all $i = 1, 2, \cdots m$. Likewise $dt_< = -\max(c_1, c_2)\delta - b/g_1$ implies $G_i(t_0 + dt_<, \vec{t_1}, \vec{t_2}) < \vartheta_2$. However, these choices for $dt$ may invalidate the condition $|dt - t_{lj}| < \delta$ for some $l, j$. Hence the Taylorexpansion is not necessarily justified for all terms in (17). This can be easily repaired by replacing $\delta$ by $\delta_2 = \delta/2$, i.e. the requirements $t_{lj} < c_l \delta_2,\ l = 1, 2$ and $(k + 2)b/g_1 < c_3 \delta_2$ (compare (6)). Then for all $l, j$ the times $t_0 + dt - t_{lj}$ fall into the neighborhood U as necessary and we can conclude that *all* cells in the target node still fire somewhere in the intervall U even for random projections. Let $t_0 + dt_{0i},\ i = 1, 2, \cdots, m$ be the respective firing times. Then, using

$$\left| \frac{1}{m} \sum_{j=1}^{m} C_{ij}^{(l)} g'(t_{ij}^{(l)}) dt_{0i} \right| \geq |dt_{0i}| g_1 \left( \frac{1}{m} \sum_{j=1}^{m} C_{ij}^{(l)} \right) \qquad \text{and} \quad (18)$$

$$\left| \frac{1}{m} \sum_{j=1}^{m} C_{ij}^{(l)} g'(t_{ij}^{(l)}) t_{lj} \right| \leq g_2 \left( \frac{1}{m} \sum_{j=1}^{m} C_{ij}^{(l)} \right) \sup_j |t_{lj}| \leq g_2 \left( \frac{1}{m} \sum_{j=1}^{m} C_{ij}^{(l)} \right) c_l \delta_2 \ . \quad (19)$$

and the fact that the lemma implies that $\lim_{m\to\infty} \sup_{i=1\cdots m} |\frac{1}{m}\sum_{j=1}^{m} C_{ij}^{(l)}| = C^{(l)} a.s.$ proceeding as in section 7.1 we find after some straightforward calculations analogous to those leading from (8) to (10) that the shifts in spike times $dt'_{1i}$ of the cells in the next firing state node are uniformly bounded by

$$\overline{\lim_{m\to\infty}} \sup_{i=1\cdots m} |dt'_{1i}| \leq \left(c_3 + \frac{c_4}{2}(c_1 + c_2)\right) \delta_2 \ . \tag{20}$$

This yields the same relation between $c_1, \cdots, c_4$ as required for the stable operation of the DFA consisting of single neurons, compare (10). The rest of the proof concerning the different kinds of input synchronization proceeds as for the automaton comprising single spiking neurons per node. Thus it follows that synfire graph networks with random projections in the asymptotic limit can stably simulate arbitrary deterministic finite state automata. Regarding the non-deterministic version of synfire graphs the results can be similarly generalized.

Because lemma 1 implies almost sure convergence, we can find for every particular automaton and every $\epsilon > 0$ a finite poolsize $M$ such that for all $m > M$ a random realization of the automaton (random with respect to the particular synaptic values) with probability $1 - \epsilon$ operates correctly on all possible input words.

**Random couplings in the finite case.** In case of finite pool sizes limit laws are not applicable. Instead we must *require* that for all cells in any given projection the condition $\left|\frac{1}{m}\sum_{j=1}^{m} w_{ij}\right| \leq \sigma$ holds for a suitable positive constant $\sigma$. The second and third term in (12) can then be bounded absolutely by $(C + \sigma)b + \sigma\hat{g} =: \tilde{b}$. Here, the first term can be made small as usual. The smallness of the second term is a constraint that we must impose on the projections. More precisely for the DFA we require that $(k+2)((C+\sigma)\hat{b}(k) + \sigma\hat{g}) = (k+2)\tilde{b} \leq c_3\delta$, for suitable $\sigma$ and $k$. $C$ can be chosen to be one and $\hat{g}$ follows from the particular choice for $g(t)$. Such a choice is possible for sufficiently small $\sigma$. With this choice the stability proof follows the same lines than before but with $b$ replaced by $\tilde{b}$. A similar constraint imposed on projections in a NFA network ensures also the stable operation of these networks.

If the sums $\left|\frac{1}{m}\sum_{j=1}^{m} w_{ij}\right|$ are not strictly bounded by $\sigma$, but may obtain larger values with a small probability, this to a certain degree might be tolerable as long as it leads only to the erroneous firing of a small fraction of cells in every pool. It is plausible to assume that a few wrong cells per pool do not harm. However, a proof of this conjecture is missing.

**Identical and diluted connections.** We should mention that the condition on $\sigma$ is trivially satisfied for identical (non-random) connections whatever the poolsize is. Moreover, in conventional synfire chain networks sometimes synfire links of the form (11) whith diluted connectivities are considered. If the nodes contain $m$ cells, only $m_1 < m$ connections might be actually realized on any cell in the target node. Obviously, if the realized synapses all have identical values, this case essentially reduces

to that of identical synapes and a complete connectivity - only the normalization in (11) has to be adapted to the multiplicities (that is $1/m$ has to be replaced by $1/m_1$ in (11).

Finally, diluted projections with random synaptic efficacies can be envisaged as a special case of general random projections as studied above. This is, because as mentioned at the beginning of this section the distribution function of the connections may have a pointmass in zero. If, for example, the connectivity of a synfire link is $p$ $(= m_1/m)$ then the distribution function has a Diracmass of weight $1-p$ in zero, which determines the fraction of vanishing connections. Note, however, that this interpretation of diluted connections is slightly different from the idea of multiplicities, because the number of actually realized synapses on a target cell now is a random variable (with expection $m_1$) and not fixed (at exactly $m_1$). Of course, a varying number of neurons projecting on different target cells should be viewed as the more plausible situation.

**Additional background noise.** So far we did not take into consideration that neurons in the network may be subject to noise as for example supplied by asynchronously firing background cells in real neural networks. Noise can be included in form of an additional random contribution to the membrane potentials. As long as the noise is bounded and small it can be treated similarly to the quenched disorder in the previous case, i.e. we may impose a bound $\sigma^*$ on the maximal fluctuations and formulate conditions on the smallness of $\sigma^*$ as above. In that case the noise sources need not even be independent. For large or unbounded noise the situation turns out to be more complicated. The problem has been discussed recently for several other automata models [8, 35, 36]. In [36] it has been shown under quite general conditions that time-discrete networks of sigmoid neurons subject to unbounded noise cannot recognize arbitrary regular languages. Although not directly comparable with our network it seems reasonable that a similar conjecture holds true in networks of spiking neurons.

The proofs above have shown that the number of delay stages for NFA networks can be chosen independently of the special automaton. In contrast for NFAs it depends on the number of states of the automaton. This implies that also the constants $\sigma$ and $\sigma^*$ in the previous paragraphs depend on the number of states. The larger a given automaton is, the stronger are the constraints imposed on these bounds. For this reason it seems unlikely that very large NFAs (of the here considered architecture) can be constructed which successfully operate under the rather unreliable conditions in the brain. Even for automata of moderate size, may they be deterministic or nondeterministic, one should expect that node firing fails with a certain probability. This suggests that perhaps probabilistic automata models with significant failure rates are more appropriate to describe computational processes going on in real brains. From this point of view deterministic computations are only a rather special limiting case.

# 8 Discussion

Frontal cortical areas, where many experiments on synfire activity are performed, participate in planing, spatial and temporal memory, and other cognitive tasks.[13, 14] A relation between the appearance of spatio-temporal spike patterns and behavioral events has been demonstrated [4, 45]. Therefore one may ask, whether synfire chains contribute to behavioral or cognitive tasks in any reasonable way. Obviously the conventional synfire chain model with linear or perhaps cyclically reverberating chains has not much explanatory power regarding mechanisms for the solution of complex tasks. Its main purpose is to provide a plausible model for the abundance of spatio-temporal spike patterns and long-time correlations. As we have shown, a rather simple extension of the model enables the construction of arbitrary deterministic and nondeterministic finite state automata. To this end it essentially suffices to introduce several possible successor nodes in any state of the chain which become selectively activated in dependence of further input signals. In this way, arbitrary 'snfire-graphs' can be built, that are able to implement any desired finite automaton. Furthermore, nondeterminism (in the sense of computer science) in this model corresponds to several simultaneously propagating waves of activation ('multiple snfire chains' [7]). We have proved formally that both versions of synfire graphs, the deterministic as well as the nondeterministic one, can reliably process arbitrary long input sequences in suitable parameter regimes. Input spike trains need not be perfectly periodic but spike times may fluctuate in certain bounds; in any case they synchronize the dynamics of the automaton. This generalizes comparable results stated by Omlin and Giles for time-discrete deterministic automata of a similar architecture, although this work uses graded neurons and multiplicative synapses [41].

Similarities of the present work also exist to the work of Maass and collaborators regarding automata of spiking neurons [33, 34]. These works investigate general bounds on the theoretical computational power of spiking neuron networks. In fact, this power turns out to be very high. For example, reference [34] proposes architectures for arbitrary threshold gates, Turing machines, and certain random access machines operating on real numbers. However, some of the assumptions that are necessary to reach this high power are probably not satisfied by real neural systems. One crucial assumption concerns the representation of real numbers necessary to go from finite automata to infinte machines like Turing machines. This representation relies on neural oscillators emitting spikes with an arbitrary precision within one period as well as relative to other neurons. Although coding by spike timing to some degree is supported experimentally, its precision is still a matter of discussion [48, 58]. Certainly it is not arbitrarily precise; but any restriction to a finite precision immediatly leads back from Turing complexity to finite automata in the architecture of Maass et al. [33, 34]. Similarly these models neglect noise and effects due to the exponential decay of postsynaptic potentials, which both restrict timing precision too. Hence, it seems that the strong results about the theoretical

power of spiking neuron networks hardly carry over to biological neural networks. In the present work we have explicitly avoided the mentioned unrealistic assumptions and included some further features probably relevant to real networks, for example, pools and diluted random projections. We have shown that our automata are robust against fluctuations in membrane potentials, that may arise from quenched disorder in synaptic efficacies, background noise, jitter in input times and the firing of cells in synfire nodes, as well as unpredictable temporal summation on membranes.

It is clearly not surprising that networks of time-continuous spiking neurons can simulate finite automata since already McCulloch and Pitts have shown that this is possible for simple time-discrete threshold neurons [38]. As for Omlin and Giles' work our results can be viewed as an extension of McCulloch-Pitts networks to spiking neurons. Per se it is not obvious that McCulloch-Pitts networks work properly when realistic single unit models and membrane properties are included. In fact our analysis implies that at least some constraints regarding minimal interspike intervals have to be satisfied, which require more than just two layers of cells. Moreover, we should also emphasize that our motivation was not so much the developement of automata as such, but a step towards a functional interpretation of synfire chains. The hypothesis is that synfire activity can perhaps be understood as an expression of computations based on neuronal micro-circuits in some way similar to our model. This hypothesis has been made concrete by the extension of the conventional synfire chain model.

# Acknowledgement

This work has been supported by DFG, Pa 268/8-1.

# References

[1] Abeles, M.: Corticonics: Neural circuits of the cerebral cortex. Cambridge University Press, Cambridge UK, 1991.

[2] Abeles, M., Bergman, H., Gat, I. Meilijson, I., Seidemann, E., Tishby, N., and Vaadia, E.: Cortical Activity Flips Among Quasi Stationary States. PNAS 92: 8616–8620, 1995.

[3] Abeles, M., Vaadia, E., Bergman, H., Prut, Y., Headman, I., and Slovin, H.: Dynamics of Neuronal Interactions in the Frontal Cortex of Behaving Monkeys. CINS 4: 131–158, 1993.

[4] Abeles, M., Bergman, H. Margalit, E., and Vaadia, E.: Spatio-temporal firing patterns in frontal cortex of behaving mokeys. J.of Neurophysiol. 70:1629–1643, 1993.

[5] Aertsen, A.; Diesmann, M., and Gewaltig, M.O.: Propagation of synchronous spiking activity in feedforward neural networks. J.Physiology(Paris) 90:243–7,1996.

[6] Arnoldi, H.M.R., Brauer, W.: Synchronization without oscillatory neurons. Biol.Cybern. 74, 209–23, 1996.

[7] Bienenstock, E.: A model of the neocortex. Network 6: 179–224, 1995.

[8] Casey, M.: The Dynamics of Discrete-Time Computation, with Application to Recurrent Neural Networks and Finite State Machine Extraction. Neural Comp. 8,1135-1178, 1996.

[9] Diesmann, M., Gewaltig, M.O., and Aertsen A.: Characterization of Synfire Activity by Propagating 'Pulse Packets'. In: Bower, J.: Computational Neuroscience: Trends in Research 1995, Chapter 10, pages 59-64. Academic Press, San Diego, 1996.

[10] Eccles, J.C.: The physiology of imagination. Scientific American, 199:135-146,1958.

[11] Eckhorn, R.; Bauer, R.; Jordan, W.; Brosch, M.; Kruse, W.; Munk, M.; Reitboeck, H.J.: Coherent oscillations: A mechanism of feature linking in the visual cortex? Biol.Cyb. 60: 121-130, 1988.

[12] Eckhorn, R.; Reitboeck, H.J.; Arndt, M.; Dicke, P.: Feature Linking via Synchronization among Distributed Assemblies: Simulations of Results from Cat Visual Cortex. Neural Comp. 2: 293-307, 1990.

[13] Fuster, J.M.: Behavioural electrophysiology of the prefrontal cortex. J. Neurophysiol. 36, 61ff, 1973.

[14] Fuster, J.M. and Jervey J.P. Inferotemporal neurons distinguish and retain behaviorally relevant features of visual stimuli, Science, 212, 952-955, 1980.

[15] Gerstner, W. and v.Hemmen, J.L.: Associative memory in a network of 'spiking' neurons. Network 3, 139–64, 1993.

[16] Gerstner, W.: Time Structure of the Activity in Neural Network Models. Phys.Rev.E 51: 738-758, 1995.

[17] Gerstner, W.; Ritz, R.; van Hemmen, J.L.: Why spikes? Hebbian learning and retrieval of time-resolved excitation patterns. Biol Cybern. 69(5-6): 503-515, 1993.

[18] Gibbon, J.; Allan, L. (eds.) : Timing and Time Perception. Special Issue of Annals of the New York Academy of Sciences, Vol.423, 1984.

[19] Gray, C.M.; König, P.; Engel, A.K.; Singer, W.: Oscillatory responses in cat visual cortex exhibit intercolumnar synchronisation which reflects global stimulus properties. Nature 338: 334-337, 1989.

[20] Gray, C.M.: Synchronous Oscillations in Neuronal Systems: Mechanisms and Functions. J.Comp. Neurosci. 1: 11-38, 1994.

[21] D.O.Hebb, The organization of behavior. A neuropsychological theory (Wiley, New York, 1949).

[22] Hertz, J., Krogh,A., and Palmer R.G.: Introduction to the theory of neural computation. Addison Wesley, 1991.

[23] Hertz, J., Prügel-Bennett, A.: Learning short synfire chains by self-organization. Network 7, 357–63, 1996.

[24] Herrmann, M., Hertz, J.A., and Prügel-Bennett, A.: Analysis of Synfire Chains. Network 6, 403–14, 1995.

[25] Hopcroft, J.E.; Ullman, J.D.: Introduction to Automata Theory, Languages, and Computation. Addison-Wesley, Reading, MA, 1979.

[26] Hopfield, J.J.: Neural Networks and Physical Systems with Emmergent Collective Computational Properties. PNAS, USA 79, 2554–8, 1982.

[27] Hopfield, J.J. Pattern recognition computing using action potential timing for stimulus representation. Nature 376:33-36,1995.

[28] Horn, D. and Usher, M.: Neural Networks with dynamical thresholds. Phys.Rev.A 40, 1036–44, 1989.

[29] Kandel,E. R. ;Schwartz,J. H. : Principles of Neural Science. Elsevier, New York, 1981.

[30] Lankheet, M.J.M., Molenaar, J., and van de Grind, W.A.: The Spike Generating Mechanism of Retinal Ganglion Cells. Vision Research 29: 505–517, 1989.

[31] Lestienne, R.; Strehler, B.L.: Time structure and stimulus dependence of precisely replicating patterns present in monkey cortical neuronal spike trains. Brain Research, 437:214-238,1987.

[32] Lukashin A.V.; Georgopoulos, A.P.: A dynamical neural network model for motor cortical activity during movement: population coding of movement trajectories. Biol. Cybern. 69:517–524,1993.

[33] Maass, W.: On the computational power of noisy spiking neurons. Advances in Neural Information Processing Systems 8:211-217. MIT-Press, Cambridge, MA.

[34] Maass, W.: Lower bounds on the Computational Power of Networks of Noisy Spiking Neurons. Neural Computation 8:1-20,1995.

[35] Maass, W.; Orponen, P.: On the Effect of Analog Noise in Discrete-Time Analog Computations. Preprint, 1996.

[36] Maass, W.; Sontag, E.D.: Analog neural nets with Gaussian or other common noise sources cannot recognize arbitrary regular languages. Preprint.

[37] von der Malsburg, C.: The correlation theory of brain function. Interner Report 81-2. Max-Planck Institut für Biophysikalische Chemie, Göttingen, 1981.

[38] McCulloch, W.S. and Pitts, W.: A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics, 5:115–133,1943.

[39] MacGregor, R.J.: Composite cortical networks of multimodal oscillators. Biol.Cybern. 69,243–255,1993.

[40] Neumann, J.von: The computer and the brain. Yale University Press. New Haven, London, 1958.

[41] Omlin, C.W. and Giles, C.L. Constructing Deterministic Finite-State Automata in Recurrent Neural Networks. Journal of the ACM, Vol.43, No. 6:937–972, 1996.

[42] Palm, G.: Neural Assemblies. An Alternative Approach to Artificial Intelligence. Springer, Berlin, 1982.

[43] Palm, G.: On associative memories. Biol.Cybern. 39, 125ff, 1981.

[44] Palm, G.: Memory Capacities of Local Rules for Synaptic Modification. CINS 2:97–128, 1991.

[45] Riehle, A.; Grün, S.; Diesmann, M.; and Aertsen, A.: Spike synchronization and rate modulation differentially involved in motor cortical function. Science 278:1950-1953, 1997.

[46] Schillen, T.B.; König, P.: Binding by temporal structure in multiple feature domains of an oscillatory neural network. Biol.Cybern. 70: 397-405, 1994.

[47] Shastri, L.; Ajjanagadde, V.: From simple associations to systematic reasoning: A connectionist representation of rules, variables and variable bindings. Behavioral and Brain Sciences 16:417-494,1993.

[48] Shadlen, M.N.; Newsome, W.T.: Noise, neural codes and cortical organization. Current Opinion in Neurobiology 4:569–579,1994.

[49] Sillito A.M, Jones H.E, Gerstein G.L, West D.C.: Feature-linked synchronization of thalamic relay cell firing induced by feedback from the visual cortex, Nature, 369:479-482,1994.

[50] Singer, W.; Gray, C.M.: Visual feature integration and the temporal correlation hypotheses. Ann.Rev.Neuroscience 18: 555-586, 1995.

[51] Stein, R.B.: Some Models of Neuronal Variability. Biophysical Journal, 7:37-68, 1967.

[52] Thorpe, S.J.: Spike arrival times: a highly efficient coding scheme for neural networks. In: Eckmiller, R.; Hartmann, G.; Hauske, G. (eds) Parallel processing in neural systems. North-Holland/Elsevier, Amsterdam, 1990, pp91–94.

[53] Tononi, G.; Sporns, O.; Edelman, G.M.: Reentry and the problem of integrating multiple cortical areas: simulation of dynamic integration in the visual system. Cerebral Cortex 2: 310-335, 1992.

[54] Villa, A.E.P. and Abeles, M.: Evidence for spatiotemporal firing patterns within the auditory thalamus of the cat. Brain Research, 509:325-327, 1990.

[55] Wennekers, T. and Palm G.: Controlling the Speed of Synfire Chains. In: von der Malsburg, C. et al. (eds.) Proceedings of the International Conference on Artificial Neural Networks ICANN96. Springer, Berlin, 1996, pp 451–56.

[56] Wennekers, T. and Palm G.: Cell Assemblies, Associative Memory and Temporal Structure in Brain Signals. In: R.Miller (Editor) Time and the Brain. Conceptual Advances in Brain Research, vol.2, Harwood Academic Publishers (presumably 1998).

[57] Wennekers, T. and Palm G.: On the relation between neural modeling and experimental neuroscience. Theory in Biosciences 116:273–289,1997.

[58] Wennekers, T. and Palm G.: How Imprecise is Neuronal Synchronization? Proc. of the Computational Neuroscience Meeting, CNS98, Santa Barbara, July 25-30, 1998.

[59] Wennekers, T.: Asymptotik rekurrenter neuronaler Netze mit zufälligen Kopplungen. Ulmer Informatik Berichte Nr. 97-15, Fakultät für Informatik, Universität Ulm, 1997.

[60] Wennekers, T.; Palm, G.: Control of the replay speed of timesequences in associative networks of spiking neurons. In preparation.

[61] Wickelgren, W.A.: Context-sensitive coding, associative memory, and serial order in (speech) behavior. Psychological Review, 76: 1-15, 1969.

[62] Wickelgren, W.: Webs, cell assemblies, and chunking in neural nets. CINS 3(1), 1–53, 1992.

[63] Willshaw, D.J.; Buneman, O.P.; and Longuet-Higgins, H.C.: Non-holographic associative memory. Nature 222:960ff,1969.

[64] Yu, A.C.; Margoliash, D.: Temporal Hierachical Control of Singing in Birds. Science 273:1871–1875,1996.