pp. 109 — 118

Self-organisation in a system of binary strings

Wolfgang Banzhaf

Department of Computer Science, Dortmund University

Baroper Str. 301, 44221 Dortmund, GERMANY

banzhaf@tarantoga.informatik.uni-dortmund.de

Abstract

We discuss a system of autocatalytic sequences of binary numbers. Sequences come in two forms, a 1-dimensional form (operands) and a 2-dimensional form (operators) that are able to react with each other. The resulting reaction network shows signs of emerging metabolisms. We discuss the general framework and examine specific interactions for a system with strings of length 4 bits. A self-maintaining network of string types and parasitic interactions are shown to exist.

Introduction

Sequences of binary numbers are the most primitive form of information storage we know today. They are able to code any kind of man-made information, be it still or moving images, sound waves and other sensory stimulations, be it written language or the rules of mathematics, just to name a few. As the success of von-Neumann computers has shown over the last 50 years, binary sequences are also sufficient to store the commands that drive the execution of computer programs. In fact, part of the success of the digital computer was due to the universality of bits and their interchangeability between data and programs.

It is not far-fetched to expect that the physical identity between operators (programs) and operands (data) may also play an essential role in self-organisation. We have proposed to consider a simple self-organising system [1], in which sequences of binary numbers are able to react with each other and sometimes even to replicate themselves. This ability of binary strings was a result of the proposition to consider binary strings similar to sequences of nucleotides in RNA. RNA sequences which presumably stood at the cradle of life [2, 3], seem capable of self-organisation and come in at least two alternative forms, a one-dimensional genotypic form and a two or three-dimensional phenotypic form. We proposed to consider binary strings in analogy and to provide for a second, folded and operative form of strings. Technically, we considered as this alternative a two-dimensional matrix form that is able to perform operations on other one-dimensional binary strings.

Reactions between binary strings

The fundamental ideas of this model have been outlined elsewhere (see ref. [1],[4],[7] for details). Here we only give a brief overview of what has been learned so far.

Let us consider sequences

$$\vec{s} = (s_1, s_2, ..., s_i, ..., s_N) \,. \tag{1}$$

of binary symbols $s_i \in \{0, 1\}, i = 1, ..., N$ organised in 1-dimensional strings.

Then we ask the question: Does there exist an alternative form of these strings, that is (i) reversibly transformable into the form (1), and is (ii) operative on form (1)? The answer is surprisingly simple and well known from mathematics: Yes, there are operators with the above capabilities, known as matrices.

Thus, we require the existence of a mapping $\mathcal{M}$

$$\mathcal{M} : \vec{s} \longrightarrow \mathcal{P}_{\vec{s}} \tag{2}$$

which transforms $\vec{s}$ into a corresponding 2-dimensional matrix form $\mathcal{P}_{\vec{s}}$ of the sequence which should be unique and reversible. This mapping is simply a spatial reorganisation of the information contained in a sequence and may be termed a *folding*, in close analogy to the notion used in molecular biology.

The most compact realization of such a 2-dimensional form would be a quadratic matrix. For a string with a quadratic number of components $N, N \in \mathcal{N}_{sq}$ with $\mathcal{N}_{sq} = \{1, 4, 9, 16, 25, ...\}$, the procedure is straightforward: Any systematic folding (examples are shown in Figure 1) would do. Since folding is not yet very sophisticated, and different configurations may be obtained by a renumbering of string components, we shall consider here the topological folding of Figure 1 (b) only.

In the more general case of N being a non-quadratic number, different generalizations are reasonable. Here

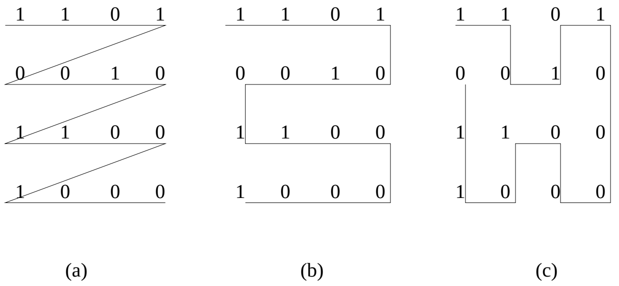

Figure 1: Some two-dimensional compactly folded forms of a string in an example with $N = 16$ binary numbers: $\vec{s} = (1101001011001000)$.
(a): non-topological folding, (b) and (c): topological foldings.

we shall only discuss a compact folding [1] in non-square matrices, where

$$N_1 \times N_2 = N. \tag{3}$$

In order to treat non-quadratic cases similar to the quadratic case, a bias should be used in the direction of the most compact solution, i.e.

$$N_i = \sqrt{N} + \varepsilon_i, \qquad i = 1, 2 \tag{4}$$

with $\mid \varepsilon_i \mid$ as small as possible.

Table 1 gives the resulting 2-dimensional form for strings up to $N = 10$. One can see that strings with a length corresponding to a prime number are somewhat special as they do not allow any compactification in the 2-dimensional form.

The interaction between a 2-dimensional form of a string and a 1-dimensional form can be considered a reaction between the two strings. As an example, let us assume an operator $\mathcal{P}_{\vec{s}}$ was formed from string $\vec{s}$. This operator might now "react" with another string, $\vec{s}\,'$, producing thereby a new string $\vec{s}\,''$:

$$\mathcal{P}_{\vec{s}}\ \vec{s}\,' \Rightarrow \vec{s}\,'' \tag{5}$$

The notion here is that some sort of raw material (analogous to energy-rich monomers in Nature) is continuously supplied to allow the ongoing production of new strings based on the information provided by the cooperation of $\mathcal{P}_{\vec{s}}$ and $\vec{s}\,'$.

A typical example of an interaction is given in Figure 2 for the simple case of strings of the same quadratic length N. $\vec{s}\,'$ might be considered as concatenated from $\sqrt{N}$ segments with length $\sqrt{N}$ each. The operator $\mathcal{P}_{\vec{s}}$ acts on each of these segments sequentially, and performs semi-local operations. In this way, it moves down the string in steps of size $\sqrt{N}$ until it has finally completed the production of a new string $\vec{s}\,''$.

Length	Compact folded form
1	(s_1)
2	$(s_1 \quad s_2)$
3	$(s_1 \quad s_2 \quad s_3)$
4	$\begin{pmatrix} s_1 & s_2 \\ s_4 & s_3 \end{pmatrix}$
5	$(s_1 \quad s_2 \quad s_3 \quad s_4 \quad s_5)$
6	$\begin{pmatrix} s_1 & s_2 & s_3 \\ s_6 & s_5 & s_4 \end{pmatrix}$
7	$(s_1 \quad s_2 \quad s_3 \quad s_4 \quad s_5 \quad s_6 \quad s_7)$
8	$\begin{pmatrix} s_1 & s_2 & s_3 & s_4 \\ s_8 & s_7 & s_6 & s_5 \end{pmatrix}$
9	$\begin{pmatrix} s_1 & s_2 & s_3 \\ s_6 & s_5 & s_4 \\ s_7 & s_8 & s_9 \end{pmatrix}$
10	$\begin{pmatrix} s_1 & s_2 & s_3 & s_4 & s_5 \\ s_{10} & s_9 & s_8 & s_7 & s_6 \end{pmatrix}$

Table 1: Compact topological string folding with length up to $N = 10$. Each folding comes also with the transposed matrix.

The particular algorithm for assembling new components "0" and "1" into strings that we have examined in more detail, is:

$$s'_{i+k\sqrt{N}} = \sigma\left[\sum_{j=1}^{j=\sqrt{N}} P_{ij}\, s_{j+k\sqrt{N}} - \Theta\right] \tag{6}$$

$$i = 1, ..., \sqrt{N} \qquad k = 0, ..., \sqrt{N} - 1$$

with $\sigma[\ ]$ being the squashing function

$$\sigma[x] = \begin{cases} 1 & \text{for } x \geq 0 \\ 0 & \text{for } x < 0 \end{cases} \tag{7}$$

and Θ used as an adjustable threshold. Eq. (6) may be interpreted as a combination of Boolean operations, applied separately in each segment k of the string if $\Theta = 1$.

The consistent generalization of eq. (6) for interaction of non-quadratic strings and for strings of different length is straightforward: Suppose a matrix of size $N_1 \times N_2$ is interacting with a string of length N_3. The operator locally interacts with N_1 elements of the second string in order to generate one component of the new string. This operation will be repeated N_2 times, then the operator moves on to interact with the next N_1 elements of the second string. The newly produced string will thus consist of N_4 elements with

$$N_4 = \lceil \frac{N_3}{N_1} \rceil \times N_2. \tag{8}$$

where $\lceil x \rceil$ are Gaussian parentheses giving the next larger integer to x.

In mathematical terms, the interaction reads:

$$s''_{i+kN_2} = \sigma\left[\sum_{j=1}^{j=N_1} P_{ij}\, s'_{j+kN_1} - \Theta\right] \tag{9}$$

[1] Compact foldings do not have any spacing between adjacent string elements

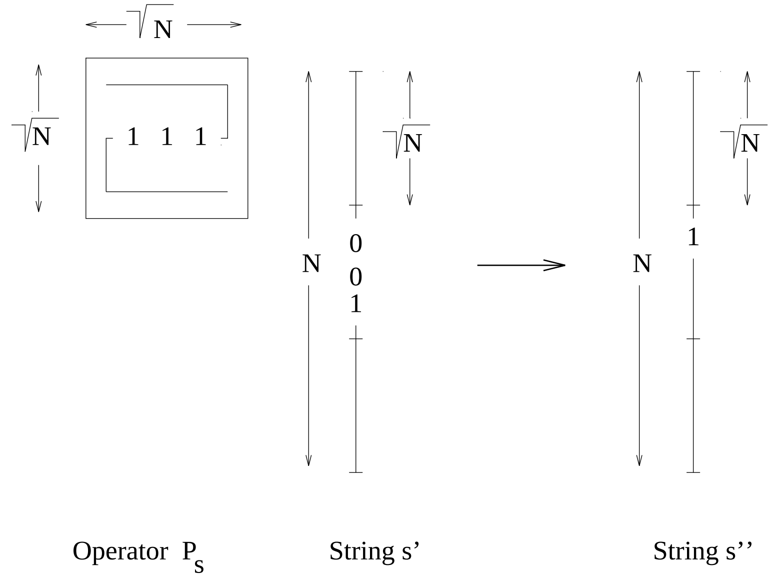

Figure 2: An operator $\mathcal{P}_{\vec{s}}$ of matrix dimension $\sqrt{N} \times \sqrt{N}$ (derived from string $\vec{s}$) acts upon a string $\vec{s}\,'$ consisting of $\sqrt{N}$ segments of length $\sqrt{N}$ each to produce a new string $\vec{s}\,''$.

with

$$i = 1, ..., N_2 \qquad k = 0, ..., \lceil\frac{N_3}{N_1}\rceil - 1.$$

This interaction is generally length-changing — either resulting in a shorter or a longer product strings. The particular direction of this length-change depends on the relation of N_1 to N_2: If $N_1 > N_2$ then $N_4 < N_3$, and the new string is shorter. If, however, $N_1 < N_2$ then $N_4 > N_3$, and the new string is longer.

The different types of possible reactions between strings are listed in Table 2. For a given N, say $N = 4$, the reactions form a reaction network, and we shall observe in the next Section the behavior of such a network.

Reactants	Product	Description
$s + s'$	s''	Heterogeneous reaction
$s + s'$	s	Replication
$s + s'$	s'	Replication
$s + s$	s'	Heterogeneous self-reaction
$s + s$	s	Self-replication

Table 2: Characterization of different polymerization reactions.

Dynamics of a sample system

Every reaction vessel is only able to keep a finite number of strings, say M. The reactions discussed in Section 2, however, continuously produce new strings. Therefore, a competitive dynamics has to be implemented by providing for an overflow mechanism for the reaction vessel. Since at present we would like a well-stirred reaction vessel without any spatial structure for reactions, the removal of strings will be a random process, hitting each sort of strings with a probability proportional to its concentration. For each newly produced string, one string is removed from the vessel. Whereas this random process does not influence in any way the constitution of the vessel content, due to different reaction channels producing new strings, a change in the composition of the content will happen over time.

There are, however, some potentially "lethal" strings in such systems. A string is said to be lethal if it is able to replicate in an unproportionally large number in almost any ensemble configuration. For eq. (6), this happens to be the case for two self-replicating string types [2], $s^{(0)} = (0, 0, ..., 0, 0)$ and $s^{(2^{N-1})} = (1, 1, ..., 1, 1)$. The former is able to replicate with every other string, the latter with most of the other strings.

In order to balance this tendency of the system we prohibit production of $s^{(0)}$ and discourage production of $s^{(2^{N-1})}$. In other words, $s^{(0)}$ will not be added to the vessel, if the reaction product should be $s^{(0)}$. Instead, a randomly selected string will be copied. We deal with $s^{(2^{N-1})}$ in a more gentle way by providing a means of non-deterministic string removal due to decay processes. The fewer the number of "1"'s a string contains, the more stable it becomes. The chance to decay therefore depends on the string feature

$$I^{(k)} = \sum_{i=1}^{N} s_i^{(k)}, \qquad k = 1, ..., M. \qquad (10)$$

[2] We shall name strings with decimal numbers corresponding to their binary sequence

$I^{(k)}$ measures the number of "1"'s in string k and determines a probability

$$p^{(k)} = (I^{(k)}/N)^n \qquad (11)$$

which determines whether a string should be removed. Usually, we set the parameter n to $n = 1$. In any case, the decay probability of $s^{(2^{N-1})}$ is 1. Once a string decays, its place might be filled
(i) with a later reaction product or
(ii) with a copy of a randomly selected string in the vessel. The latter method has the advantage of allowing a constant string number M in the vessel and is adopted here.

One sweep through the algorithms hence consists of the following steps:

STEP 1:

Generate M random binary strings of length N each

STEP 2:

Select a string and fold it into an operator by forming a compact matrix

STEP 3:

Select another string and apply the operator generated in STEP 2

STEP 4:

Release the new string, the old string and the operator (as string) into the reaction vessel, provided it is not an $s^{(0)}$. Otherwise go to STEP 2.

STEP 5:

Remove one randomly chosen string in order to compensate for the addition of a string in STEP 4

STEP 6:

Select one string and substitute it according to the probability of (11) with the copy of a randomly selected string

STEP 7:

Go to STEP 2

M sweeps through this algorithm are called a generation.

For a discussion of the system's dynamic behaviour we use as observables the concentrations $x_i(t)$ of all the different string types $s^{(i)}$ with:

$$x_i(t) = m_i(t)/M \qquad (12)$$

where $m_i(t)$ is the number of actual appearances of string type $s^{(i)}$ in the vessel at time t.

If we run a system by seeding it with an initial composition of M random strings, we regularly observe a transition into a (mostly fixed point) attractor. Due to different rates of production of different sorts, an initial composition will change until an equilibrium is reached. During the transition, new sorts are produced, already present sorts disappear, and every now and then a coexistence between sorts is reached for some time. As long as new sorts are created by interactions between already present sorts, the network has to reorganise itself in order to incorporate the newly emerging reaction channels between the different sorts. After some time, however, no new string sorts arrive, and the system reaches a steady state. Thus, the system behaves as one of the metabolic networks that are discussed in Bagley et. al. [5, 6]. As long as we have a small number of sorts, we can easily describe the system by a set of deterministic differential equations for the time development of string sort concentrations.

Deterministic rate equations were derived in [1] and are given here as a summary: We assume continuous non-random concentration functions $y_i(t)$ of the different string types $i, 1 \leq i \leq n_S$, which are considered to approximate the time averaged concentrations $<x_i>_t$:

$$y_i(t) \cong <x_i>_t, \qquad 0 \leq y_i(t) \leq 1 \qquad (13)$$

The deterministic rate equations in $y_i(t)$ read:

$$\dot{y}_i(t) = A(t)y_i(t) + \left[B_i y_i(t) + \sum_{k\neq i}^{n_S} C_{ik} y_k(t) - D_i\right] y_i(t) +$$

$$\textstyle\sum_{j,k\neq i}^{n_S} W_{ijk} y_j(t) y_k(t) - \frac{y_i(t)}{\sum_k y_k(t)}\Phi(t) \qquad (14)$$

where B_i, C_{ik}, W_{ijk} are coupling constants derived from a reaction table containing all sorts $1...n_S$. D_i determines a selection term

$$D_i = p^{(i)} \qquad (15)$$

and $A(t)$ reflects the addition of strings due to random copies

$$A(t) = \sum_{i,j}^{n_S} a_{ij} y_i(t) y_j(t) + \sum_{i}^{n_S} D_i y_i(t) \qquad (16)$$

where

$$a_{ij} = \begin{cases} 1 & \text{if the reaction of } s^{(i)} \text{ and } s^{(j)} \text{ produces } s^{(0)} \\ 0 & \text{otherwise .} \end{cases} \qquad (17)$$

Finally, $\Phi(t)$ is a flow term that enacts competition between the various string sorts $s^{(i)}$ by enforcing constancy of the overall sum of concentrations.

The reaction table listing the interactions between string types (cf. Table 3) can be used to derive interaction graphs for various situations. In Figure 3 we have depicted all interaction graphs that can be generated from Table 3 if we start the reaction vessel with one out of 2^{N-1} string types (here $N = 4$). Functionally identical graphs are not depicted. Figure 3 illustrates the variety of interactions emerging from a start with different string types. It ranges from self-replication over parasitic interaction to entire metabolisms. From an interaction graph it is evident, what kind of attractor may be approached.

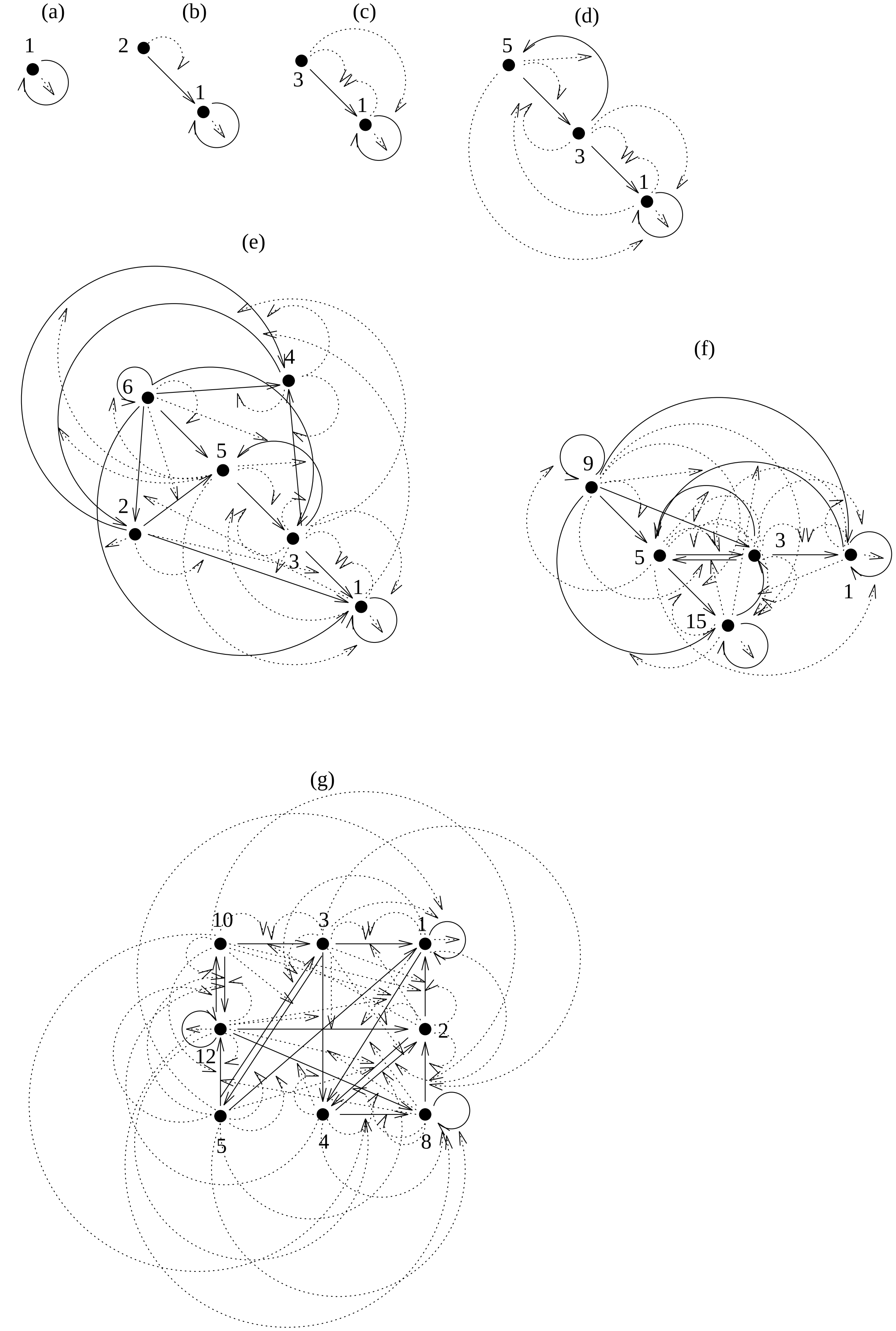

Figure 3: Interaction graphs of the system with $N = 4$. These graphs include all string sorts that are produced if the upper left string sort is used as the one and only initial sort in the reaction vessel. Solid lines connect the two string sorts participating in a reaction. Dashed lines indicate operator sort. (a) Self-replicator, also realized by sorts $s^{(7)}, s^{(12)}$ and $s^{(15)}$. (b) - (f): Simple and complicated parasitic interactions, (b) also realized by pairs $(s^{(11)}, s^{(15)}), (s^{(13)}, s^{(15)})$ and $(s^{(14)}, s^{(15)})$.

	String														
Operator	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	1	2	3	2	3	0	1	0	1	2	3	2	3
2	0	1	1	0	0	1	1	2	2	3	3	2	2	3	3
3	1	1	1	2	3	3	3	2	3	3	3	2	3	3	3
4	0	4	4	0	0	4	4	8	8	12	12	8	8	12	12
5	1	4	5	2	3	6	7	8	9	12	13	10	11	14	15
6	0	5	5	0	0	5	5	10	10	15	15	10	10	15	15
7	1	5	5	2	3	7	7	10	11	15	15	10	11	15	15
8	4	0	4	8	12	8	12	0	4	0	4	8	12	8	12
9	5	0	5	10	15	10	15	0	5	0	5	10	15	10	15
10	4	1	5	8	12	9	13	2	6	3	7	10	14	11	15
11	5	1	5	10	15	11	15	2	7	3	7	10	15	11	15
12	4	4	4	8	12	12	12	8	12	12	12	8	12	12	12
13	5	4	5	10	15	14	15	8	13	12	13	10	15	14	15
14	4	5	5	8	12	13	13	10	14	15	15	10	14	15	15
15	5	5	5	10	15	15	15	10	15	15	15	10	15	15	15

Table 3: Reactions table for the simulations of a $N = 4$ system. It was generated using a variant of (6) with topological folding.

The dynamics of the parasitic interactions of Figure 3 is examined by integrating eq. (14). Figure 4 - 6 show the results of a simulation. The transition of the string composition is clearly visible. In [1, 4] we have shown that simulations on the reaction level agree completely with the integration of rate equations used here.

A simple metabolism emerges if we do not start with one sort only, but with two or more from the outset. Figure 7 shows the interaction graph of this self-maintaining network of reactions. This graph is somewhat special as each reaction channel is of nearly equal strength. A search through the space of all combinations of 2 initial sorts uncovers that the self-replicator $s^{(12)}$ plays some special role. Usually, as soon as even a spurious concentration of $s^{(12)}$ is present, together with one other sort (except $s^{(1)}$), the metabolic attractor emerges. Figure 8 gives two examples.

It is interesting to note that there are many closed subsets of elements within even a simple $N = 4$ system. In Tables 4, 5 we give a complete list of them, ordered according to their complexity in terms of participating string sorts. Following [8], a closed subset is defined as the set $\mathcal{A}^*$ of elements from the ensemble of string types $\mathcal{N}_S = \{s^{(1)}, s^{(2)}, ..., s^{(2^{N-1})}\}$,

$$\mathcal{A}^* \subseteq \mathcal{N}_S \tag{18}$$

that might be produced by all different sequences of n reactions,

$$R_n(\mathcal{A}) = \cup_{i=0}^{n} r_n(\mathcal{A}) \tag{19}$$

starting from an initial set $\mathcal{A} \subseteq \mathcal{N}_S$, for $n \rightarrow \infty$, with

$$r_0(\mathcal{A}) = \mathcal{A} \tag{20}$$

$$r_n(\mathcal{A}) = \cup_{i=0}^{n-1} r_i(\mathcal{A}) \circ r_{n-i-1}(\mathcal{A}) : \tag{21}$$

$$\mathcal{A}^* = \lim_{n \rightarrow \infty} R_n(\mathcal{A}). \tag{22}$$

Closed subsets are important organisational structures, especially in the light of the fact, that we can only populate part of sequence space, once the component number strings increases.

We should keep in mind, that we have dealt here with a system consisting of 4-bit components. The complexity of interactions in such a simple system as Figure 3 demonstrates, is astonishing. We expect the two basic behavioral classes, parasitic interaction and metabolism, to emerge in a variety of forms in systems with longer strings.

Evolution

As we have seen, the dynamics in this small system quickly settles into one of its attractor states. The question, however, arises, whether there is a perspective for evolution, that is, for a sequential exploration of possibilities. For evolution to happen, an occasional mutation of one string into another should lead to a cascade of newly

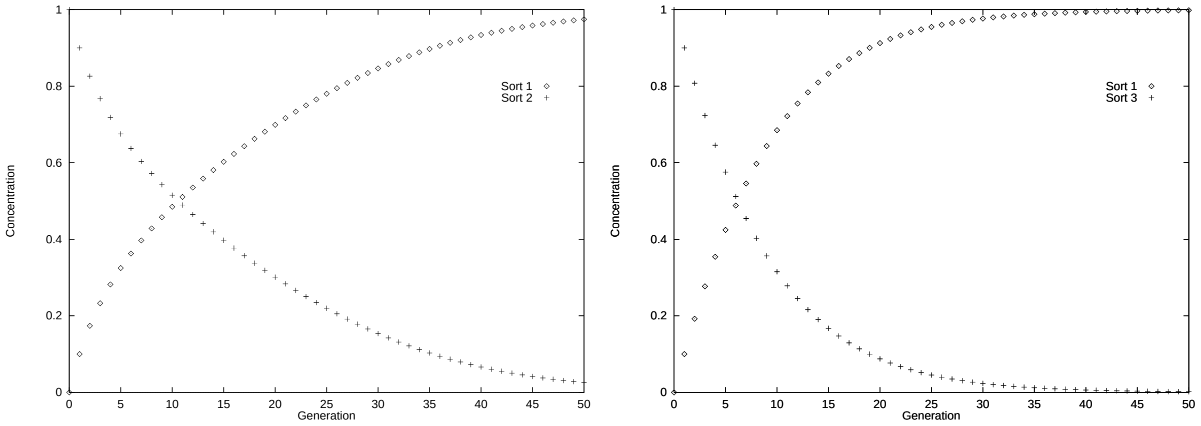

Figure 4: Dynamics of the interaction graph Figure 3(b) and (c).

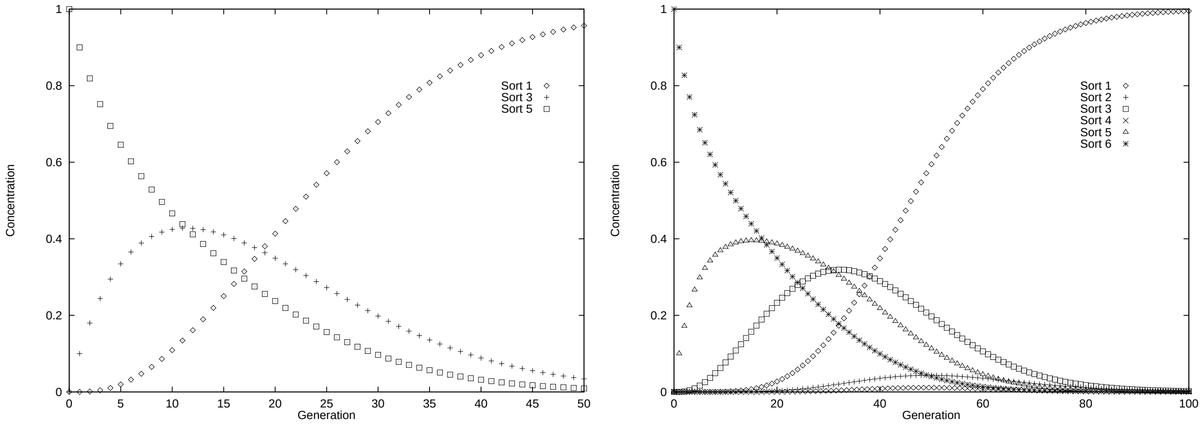

Figure 5: Dynamics of the interaction graph Figure 3(d) and (e).

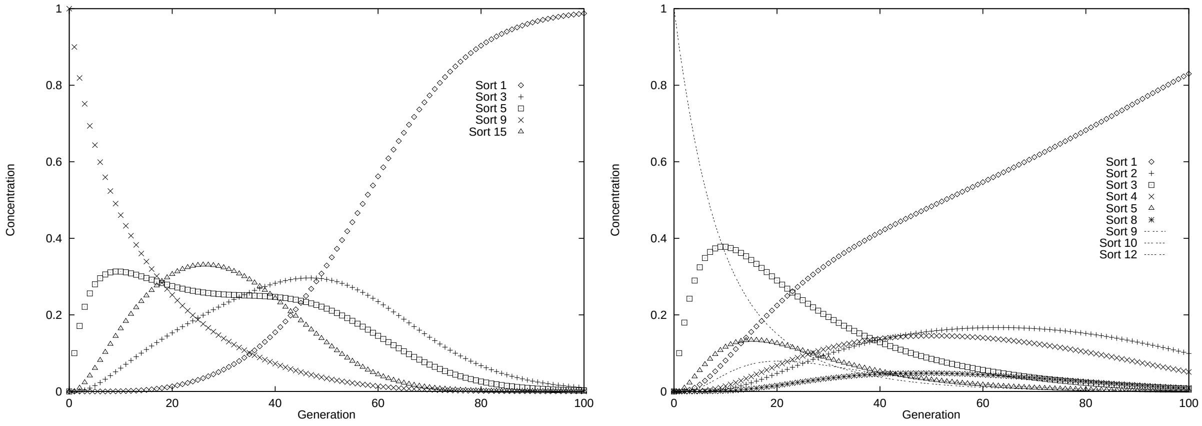

Figure 6: Dynamics of the interaction graph Figure 3(f) and (g).

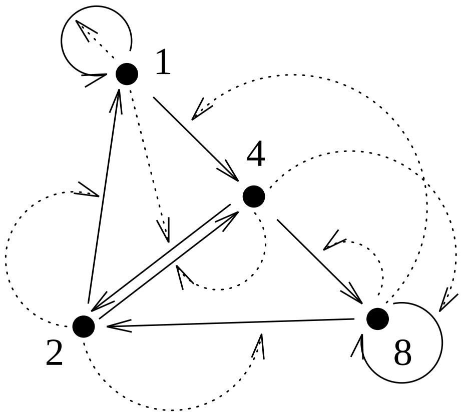

Figure 7: Reaction graph of the metabolism of $N = 4$.

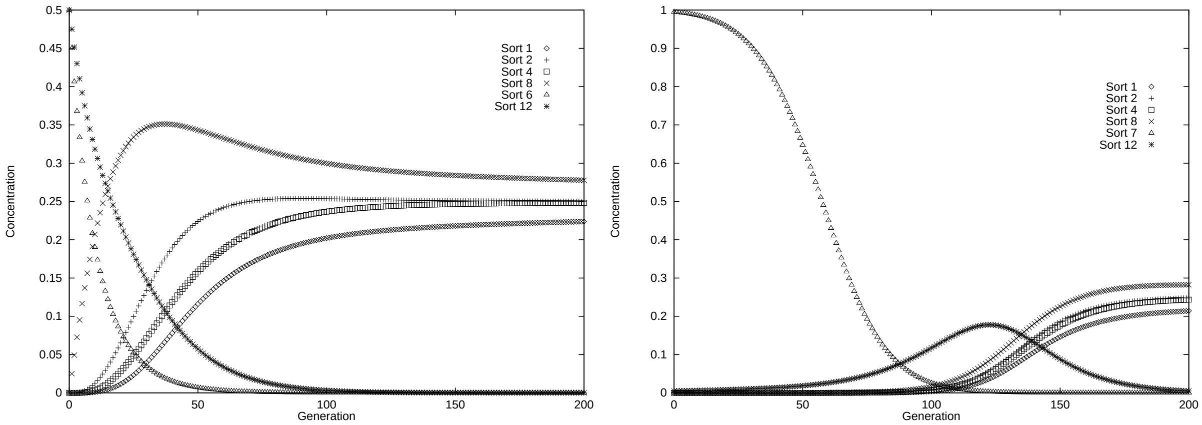

Figure 8: Dynamics of the interaction graph Figure 7. Left: Equal concentration of $s^{(6)}$ and $s^{(12)}$, at the outset; Right: High concentration of $s^{(7)}$, low concentration of $s^{(12)}$ at the outset.

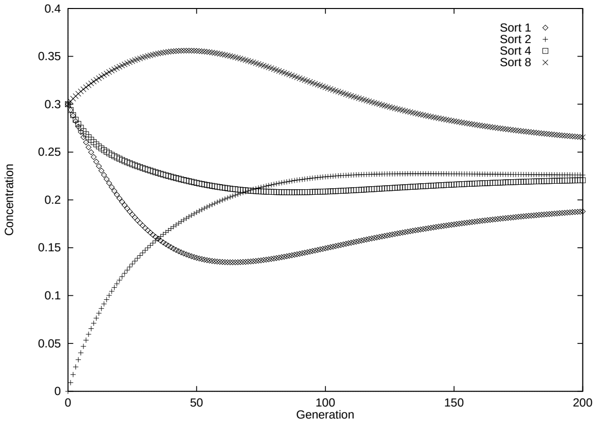

Figure 9: Dynamics of the interaction graph Figure 7. Starting with three different string sorts.

1	2	3	4	5	6	7	8
1	(1,2)	(1,2,3)	(1,2,3,4)	(1,2,3,4,5)	(1,2,3,4,5,6)	(1,2,3,4,5,6,7)	(1,2,3,4,5,8,10,12)
(4)	1,3	(1,2,4)	(1,2,4,8)	(1,2,3,4,8)	(1,2,3,4,5,7)	1,3,5,7,11,13,15	1,3,5,7,9,11,13,15
7	(4,8)	1,3,5	1,3,5,7	(1,2,4,8,12)	(1,2,3,4,8,12)		
(8)	7,15	(4,8,12)	1,3,5,15	1,3,5,7,15			
15	(8,12)	7,11,15	7,11,13,15	1,3,5,9,15			
	13,15	13,14,15		7,11,13,14,15			
	14,15						

Table 4: Closed subsets of elements with up to 8 members. First column: Self-replicators. In parenthesis: Subsets which occasionally produce the destructor.

9	11	13	15
(1,2,3,4,5,8,10,12,15)	(1,2,3,4,5,6,8,9,10,12,15)	(1,2,3,4,5,7,8,10,11,12,13,14,15)	(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15)

Table 5: Closed subsets with more than 8 members. All subsets occasionally produce the destructor.

produced string types, that lead to a new equilibrium. We have shown this to happen in a $N = 9$ system [7], and will adopt the results learned there.

We have been using a mutation as a motor for occasional change. A mutation hits each string with a probability depending on its size. We define q to be the probability that one element of a randomly selected string changes to another symbol, here "0" to "1" and vice versa. Since each element may be hit, this is a length dependent change and the probability that at least one error occurs in a string is $Q(1) = Nq$, with the provision that $q << \frac{1}{N}$. Evidently, this mutation probability depends linearly on the concentration of string sorts in the reaction vessel. That is to say, a more successful string sort will spawn more variations. Two-bit mutations are then $Q(2) = (Nq)^2$ where we neglect the fact that sometimes back-mutation may happen. In Nature, at least on instance of this type of mutations occurs in mutations caused by cosmic radiation.

Mutation does open up new transformation pathways between string sort, something Bagley et al. term a stochastic metadynamics [6].

Suppose we start our system by sort $s^{(7)}$. Since this is a self-replicating string sort, nothing interesting will happen, unless the mutation process introduces one of its nearest neighbors $s^{(3)}, s^{(5)}, s^{(6)}, s^{(15)}$. The reaction table shows, that the appearance of $s^{(6)}$ will have no consequence, whereas the appearance of $s^{(3)}, s^{(5)}, s^{(15)}$ allows the system to switch to another attractor. Figure 10, left, shows the effect of introducing $s^{(5)}$. As a result, the interaction graph of Figure 3 (d) comes into play, and $s^{(1)}$ dominates. Figure 10,right, is the evolution from selfreplicator $s^{(15)}$ to the metabolism consisting of $s^{(1)}, s^{(2)}, s^{(4)}, s^{(8)}$. This has been achieved by introducing $s^{(12)}$, a two-bit mutation from $s^{(15)}$, in spurious concentration.

Conclusion

We have examined a very simple self-organising system. The main idea was to introduce a second form of the information carriers of our system, the sequences of binary numbers. This has been accomplished by using an operative matrix form for the strings. We then have defined a particular interaction between matrices and strings and considered the interaction itself as some sort of a reaction with input and output. The low-level ("atomic") computations in the system have thus been likened to chemical reactions in the real world.

It has been shown that closed subsets of strings exist which can be considered as organisations. Under the assumption of one particular folding, these subsets of strings might be studied in their 2-dimensional matrix form alone, effectively yielding an interesting class of mathematical objects that are closed under the proposed non-linear interaction.

We also dealt with the dynamics of the competitive system naturally emerging, with reactions going on between different species of strings. As in other artificial systems [5, 6, 9, 10, 11] an attractor state was reached relatively quickly, beyond which nothing interesting happened any more. However, we already demonstrated powerful evolutionary effects brought about by the inclusion of a mutation or the potential of length changing interactions. Systems with longer strings will certainly possess different metabolic networks, and it is clear that the behavioral flexibility in such systems will be enormous.

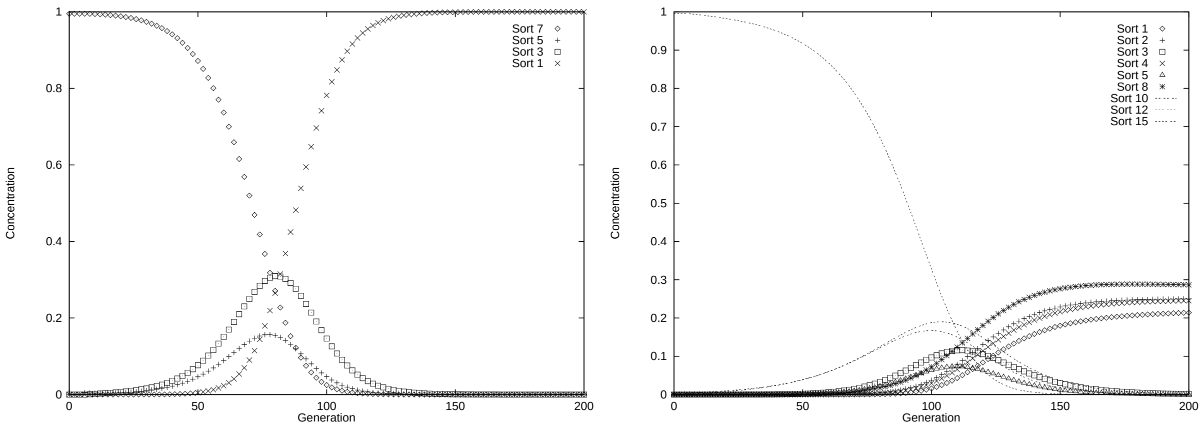

Figure 10: Evolutionary dynamics. Left: A 1-bit mutation causes $s^{(5)}$ to appear. This leads to $s^{(1)}$ as the dominant string sort after the transition. Right: A 2-bit mutation causes $s^{(12)}$ to appear. The result is the emergence of the metabolism of Figure 7.

Acknowledgement

I have enjoyed discussions with Dr. Walter Fontana. Mr. Helge Baier has provided me with the data of Table 4 and 5.

References

[1] W. Banzhaf, *Self-replicating sequences of binary numbers*, Computers and Mathematics, **26** (1993) 1

[2] M. Eigen, *Steps toward Life: a perspective on evolution*, Oxford University Press, 1992

[3] M. Eigen, P. Schuster, *The Hypercycle - A principle of natural self-organization, Part A-C*, Naturwissenschaften **64** (1977) 541 and **65** (1978) 7, 341

[4] W. Banzhaf, *Self-replicating sequences of binary numbers — Foundations I and II*, Biological Cybernetics, **69** (1993) 269 and 275

[5] R.J. Bagley, J.D. Farmer, *Spontaneous Emergence of a Metabolism*, in: C.G. Langton, C. Taylor, J.D.Farmer, S. Rasmussen (Eds.), *Artificial Life II.* Addison-Wesley, Redwood City, CA, 1991, 93

[6] R.J. Bagley, J.D. Farmer, W. Fontana, *Evolution of a Metabolism*, in: C.G. Langton, C. Taylor, J.D.Farmer, S. Rasmussen (Eds.), *Artificial Life II.* Addison-Wesley, Redwood City, CA, 1991, 141

[7] W. Banzhaf, *Self-replicating sequences of binary numbers — Foundations III*, to be published

[8] W. Fontana, L. Buss, *"The arrival of the Fittest": Toward a Theory of Biological Organization*, Bulletin of Mathematical Biology, submitted

[9] W. Fontana, *Algorithmic Chemistry*, in: C.G. Langton, C. Taylor, J.D.Farmer, S. Rasmussen (Eds.), *Artificial Life II.* Addison-Wesley, Redwood City, CA, 1991, 159

[10] J.D. Farmer, S.A. Kauffman, N.H. Packard, *Autocatalytic Replication of Polymers*, Physica **D22** (1986) 50

[11] S. Rasmussen, C. Knudsen, R. Feldberg, *Dynamics of Programmable Matter*, in: C.G. Langton, C. Taylor, J.D.Farmer, S. Rasmussen (Eds.), *Artificial Life II.* Addison-Wesley, Redwood City, CA, 1991, 211