Selforganization in a system of binary strings with topological interactions

W. Banzhaf [a,1] P. Dittrich [a] and B. Eller [a,2]

[a] *Dept. of Computer Science*
Universität Dortmund,
44221 Dortmund, GERMANY

We consider an artificial reaction system whose components are binary strings. Upon encounter, two binary strings produce a third string which competes for storage space with the originators. String types or species can only survive when produced in sufficient numbers. Spatial interactions through introduction of a topology and rules for distance-dependent reactions are discussed. We observe three different kinds of survival strategies of binary strings: cooperation of strings, exploitation of niches and parasitic behavior.

1 Introduction

The physical identity between programs and data in a von-Neumann computer provides a proliferous medium for the study of self-organization phenomena. In a von-Neumann computer, it is only a matter of interpretation whether a certain string holds numbers or characters to be manipulated or, whether the string is in fact a sequence of instructions that can be used to manipulate other strings in a particular manner. Self-organization comes about when the same strings are used interchangeably as data and programs, or, in more mathematical terms as operands and operators.

It is interesting to note that many natural systems showing self-organization are in fact based on the same principle [1–3]. Even the most complicated of them all, Life, is believed to be the result of self-organizing processes. In recent years interest has grown in the study of life-like behavior in artificial systems [4–8]. The motivation has been that researchers hope to destill principles that

[1] Presently at: International Computer Science Institute, Berkeley, CA, 94708
[2] Now at: Victoria Versicherungen, 40198 Düsseldorf, GERMANY

Preprint submitted to Elsevier Preprint *11 November 1997*

are necessarily leading to complex phenomena like Life without, however, being forced to copy nature to a large extent or in great detail.

A simpler variant of the binary string system we consider here has been proposed earlier [9] to serve as a study ground for phenomena of selforganization. It was inspired by pre-biotic evolution and the so-called RNA world [10,11]. RNA is a macromolecule that does not only serve as a storage device for biological information (like DNA, if only less stable), but that also shows some biological activity (like proteins, if only weaker). In other words, RNA has precisely this property of being able to serve (at times) as operator and (at other times) as operand. The only "trick" nature uses to achieve that is to fold strings of ribo-nucleotides, the linear sequence of which can be interpreted as information, into a two- and sometimes three-dimensional form that can act on other RNA strings and sometimes even on itself [12,13].

Thus we consider a population of binary strings (holding information) which come in an alternative, folded form as two-dimensional matrices. This second form is able to operate on strings and this operation can be used to produce new strings. The application of an operator from the population to a string from the population hence increases the number of strings and, given a finite amount of strings allowed in the population, results in a competition that imposes a selection pressure for more prolific string types [3]. It can be said that the system, through this selection pressure, organizes itself. By observing the constitution of the string population, from a possible random beginning, we can follow this process very closely and study emergent phenomena like cooperation between networks of strings.

Self-organization, as we understand it here, is the spontaneous emergence of cooperation among the elements of a system. It is spontaneous in the sense that there is no direct external influence "programming" the cooperation [14]. So far the variants of the system we have studied had made use of a random determination of operators and operands for the production of new strings. Each time a new string was inserted into the population, a randomly selected old string was destroyed. In this contribution we want to report on results with a system that has an additional topological structure imposed on the string population. Thus, each string is attached to a place, here, on a 2-dimensional grid. Reactions preferably take place with strings from the neighborhood. As will be seen in the following sections this gives rise to very interesting phenomena – among them self-sustaining spatial structures – that could not be observed without spatial interactions.

The remainder of this paper is organized as follows: Section 2 will formally summarize the original binary string system. Section 3 will introduce the

[3] We shall use the terms "type", "sort" and "species" interchangeably.

variant with topological structure used here. We shall introduce a quantity parametrizing the neighborhood such that the original system follows as a special case. Section 4 will then discuss survival in niches. Section 5 is intended to discuss the phenomenon of metabolism, as we observe it in this system. Section 6, finally contrasts present results with results in other Alife systems and draws some conclusions.

2 The binary string system without topology (the well-stirred case)

The original system was introduced in [9] and subsequently discussed in more detail in [15–18]. We shall discuss it here in a variant suitable for the following sections.

Binary strings of length N as used here are defined as follows:

$$s = (s_1, s_2, \ldots, s_N), \;\; s_i \in \mathbb{B} = \{0, 1\}, \;\; 1 \leq i \leq N, \;\; N \in \{n^2 \mid n \in \mathbb{N}\}.$$

The number of different sorts of binary strings in a population is $|\mathbb{B}^N| = 2^N$. Usually, we simplify notation by writing $s^{(k)}$ for a string species where $0 \leq k \leq 2^N - 1$ and k is determined by

$$k = \sum_{i=1}^{N} s_i \, 2^{i-1}.$$

The folding of strings of length N, with N a square number, into matrix operators $\mathbf{O}$ is achieved by the following mapping:

$$\mathcal{F} \; : \; \left\{ \begin{array}{ccc} \mathbb{B}^N & \longrightarrow & \mathbb{B}^{(\sqrt{N}, \sqrt{N})} \\ & & \\ (s_1, \ldots, s_N) & \longmapsto & \begin{pmatrix} s_{\sigma(1)} & s_{\sigma(2)} & \cdots & s_{\sigma(\sqrt{N})} \\ s_{\sigma(\sqrt{N}+1)} & s_{\sigma(\sqrt{N}+2)} & \cdots & s_{\sigma(2\sqrt{N})} \\ \vdots & \vdots & \ddots & \vdots \\ s_{\sigma(N-\sqrt{N}+1)} & s_{\sigma(N-\sqrt{N}+2)} & \cdots & s_{\sigma(N)} \end{pmatrix} \end{array} \right.$$

This mapping is bijective and was generalized to strings of arbitrary length in [19].

The effect of an operator upon a string usually results in the production of a new string. The new string is defined by the "reaction function" $\otimes$:

$$\otimes : \mathbb{B}^{(\sqrt{N},\sqrt{N})} \times \mathbb{B}^N \to \mathbb{B}^N.$$

that takes an operator-matrix $\mathbf{O} = \mathcal{F}(s) \in \mathbb{B}^{(\sqrt{N},\sqrt{N})}$ and a string $s' \in \mathbb{B}^N$, and produces a new string $s'' = (s''_1, \ldots, s''_N)$ according to the following rule:

$$s'' = \mathbf{O} \otimes s'$$

with

$$s''_{i+k\sqrt{N}} \stackrel{\text{def}}{=} \sigma\left(\sum_{j=1}^{\sqrt{N}} o_{ij}\, s'_{j+k\sqrt{N}} - \Theta\right), \quad i = 1, \ldots, \sqrt{N}, \;\; k = 0, \ldots, \sqrt{N} - 1.$$

Here, $\Theta \in \{0, \ldots, \sqrt{N} + 1\}$ is an appropriate threshold (often $\Theta = 1$), and σ is the step function:

$$\sigma(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases}$$

As we can see, this reaction function maps segments of strings from s', using $\mathbf{O}$ into segments of s''. One string type, however, poses a particular problem for the algorithm: the string consisting of 0s only. It is treated as an exception from the rule, effectively disappearing from the system should it have been present at the outset.

We now summarize the algorithm in a version that will be used later on after we have introduced topology to the system.

(i) Randomly generate a population of M binary strings of length N each.
(ii) Select an arbitrary string s from the population. Determine its form as an operator-matrix $\mathbf{O} = \mathcal{F}(s)$.
(iii) Select a second arbitrary string from the population. Compute $s'' = \mathcal{F}(s) \otimes s'$.
(iv) Put s, s' back into the population. If s'' is not an exception string, add s'' to the population and remove an arbitrary string from the population, in order to balance the total number of strings.
(v) Go to step ii.

An iteration of the algorithm (steps ii-iv) can be interpreted as a collision of two strings. If an exception occurs, no reaction product is inserted into the population. In this case the collision is called **ellastic**. If s'' is added to the

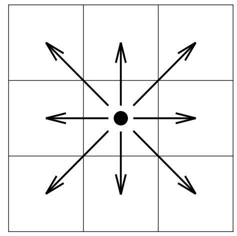

Fig. 1. Moore neighborhood: Each of the eight neighbors of the operator is selected with equal probability.

population the collision is called **reactive**. A **replication** takes place if $s'' = s$ or $s'' = s'$. A string s is called **self-replicator** if $s = \mathcal{F}(s) \otimes s$ holds.

3 The binary string system with 2-dimensional topology

The 2-dimensional topology we consider here is a simple perpendicular grid structure where on each point of the grid, one string is located. String interaction is restricted to suitably defined neighborhoods, in contrast to the well-stirred case considered in the last section. There, each string had the same probability of interaction with every other string.

In this contribution, the algorithm is modified in that, after the operator string has been selected (step i), the string it is allowed to act on is taken from only a neighborhood of the location of the operator string. Thus, once the region of the reaction has been determined by selecting a first string, the second follows from a certain area around. In terms of reactions, a molecule is much more probable to react with a neighboring molecule than with a far distant one.

The two topologies we consider are (1) the Moore neighborhood of nearest neighbors of Figure 1 and (2) the Gaussian neighborhood of Figure 2. The distance from the center in the Gaussian neighborhood is computed with a Gauss distribution. Let z be normally distributed

$$z = \mathcal{N}(0, d)$$

with center 0 and width d, then a reads

$$a = |z| + \frac{1}{\sqrt{2}}$$

where d is a parameters for the width of the Gaussian distribution. Here, a grid distance of unit length is assumed and the constant factor helps to locate a outside of the center cell. We chose periodic boundary conditions.

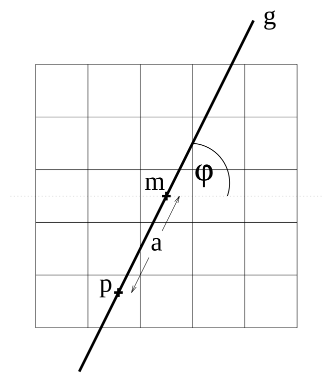

Fig. 2. Gaussian neighborhood: A point is selected according to its angle $\varphi \in [0, 2\pi[$ (distributed uniformly) and its distance a from the center.

The parameters that stay fixed in the system are:

- Length of strings $N = 4$ (resulting in 15 string species without $s^{(0)}$)
- Size of population $M = 10,000$
- Grid of 100×100 points, where strings are located
- Replacement policy: The resulting new string is placed at the point of the former operand.
- Exception handling: Reactions producing string $s^{(0)}$ are disabled.

The low number of different string types in such a system helps to avoid artifacts due to string types going extinct simply by accident. Indeed, the average number of strings of any given sort is 667, greatly reducing the probability of spurious effects.

Two folding algorithms $\mathcal{F}$ are considered: topological folding and non-topological folding. Following [15], a folding is called "topological" if neighboring string elements in the string sequence remain neighbors after the folding. If this condition is not met, we call the folding non-topological. Here we use:

$$\mathcal{F}_t : (s_1, s_2, s_3, s_4) \mapsto \begin{pmatrix} s_1 & s_2 \\ s_4 & s_3 \end{pmatrix}$$

$$\mathcal{F}_{nt} : (s_1, s_2, s_3, s_4) \mapsto \begin{pmatrix} s_1 & s_2 \\ s_3 & s_4 \end{pmatrix}$$

As a result, each string is assigned a folded operator. On encountering a string in its neighborhood, it will produce the string which the reaction function prescribed. Tables 1 and 2 show, for convenience, which reactions will take place when operator strings encounter operand strings. It should be noted that the

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

Table 1
Reaction table for topological folding.

reaction tables shown here are not providing for random interactions between strings as is discussed by, e.g., [20]. A comparison to random interactions would be interesting is beyond the scope of this paper.

Before looking at the actual dynamics of the system, an initial state should be assigned to the system. Two different conditions were studied: (a) a homogeneous initial state, i.e. the string for each point on the grid is assigned by selecting randomly from the set $\{s^{(1)}, s^{(2)}, \ldots, s^{(15)}\}$, or, alternatively, an inhomogeneous initial state, i.e. only a subset of all string sorts is used to seed the system, or the distribution of string sorts in the various regions of the grid is non-uniform. Either case will lead to interesting phenomena to be discussed later.

We now summarize the applied algorithm again:

(i) Generate an initial distribution of strings on the grid.
(ii) Select an arbitrary point c with its string s.
Fold the string into an operator-matrix by calculating $\mathcal{F}(s)$.
(iii) Select, according to the neighborhood distribution a new point c' with string s'. Calculate $s'' = \mathcal{F}(s) \otimes s'$.
(iv) If s'' is not $s^{(0)}$, replace s' at location c' by s''.
(v) Go to step (ii).

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

Table 2
Reaction table for non-topological folding.

An iteration of steps (ii-v) is called a **collision**. Time is measured by counting the number of collisions. A collision might be **reactive** if s'' is inserted in step (iv) or **elastic** otherwise.

4 Survival in spatial niches

We now want to demonstrate that the system with topology is a generalization of the system without. In Figure 3 one can see average concentrations of string types for three different systems, (i) Moore neighborhood; (ii) Gaussian neighborhood; (iii) without topology using the topological folding method.

4.1 Homogeneous initial state

We can see that for about $5 \cdot 10^5$ reactions, concentrations are changing. After that obviously an attractor has been reached causing the system to stagnate. It is interesting to note that there is a transition from very small neighborhoods to an infinite neighborhood in terms of the differentiation of string concentration levels. The smaller the neighborhood, the more differentiated the levels of concentration for different species become. One can also observe very clearly

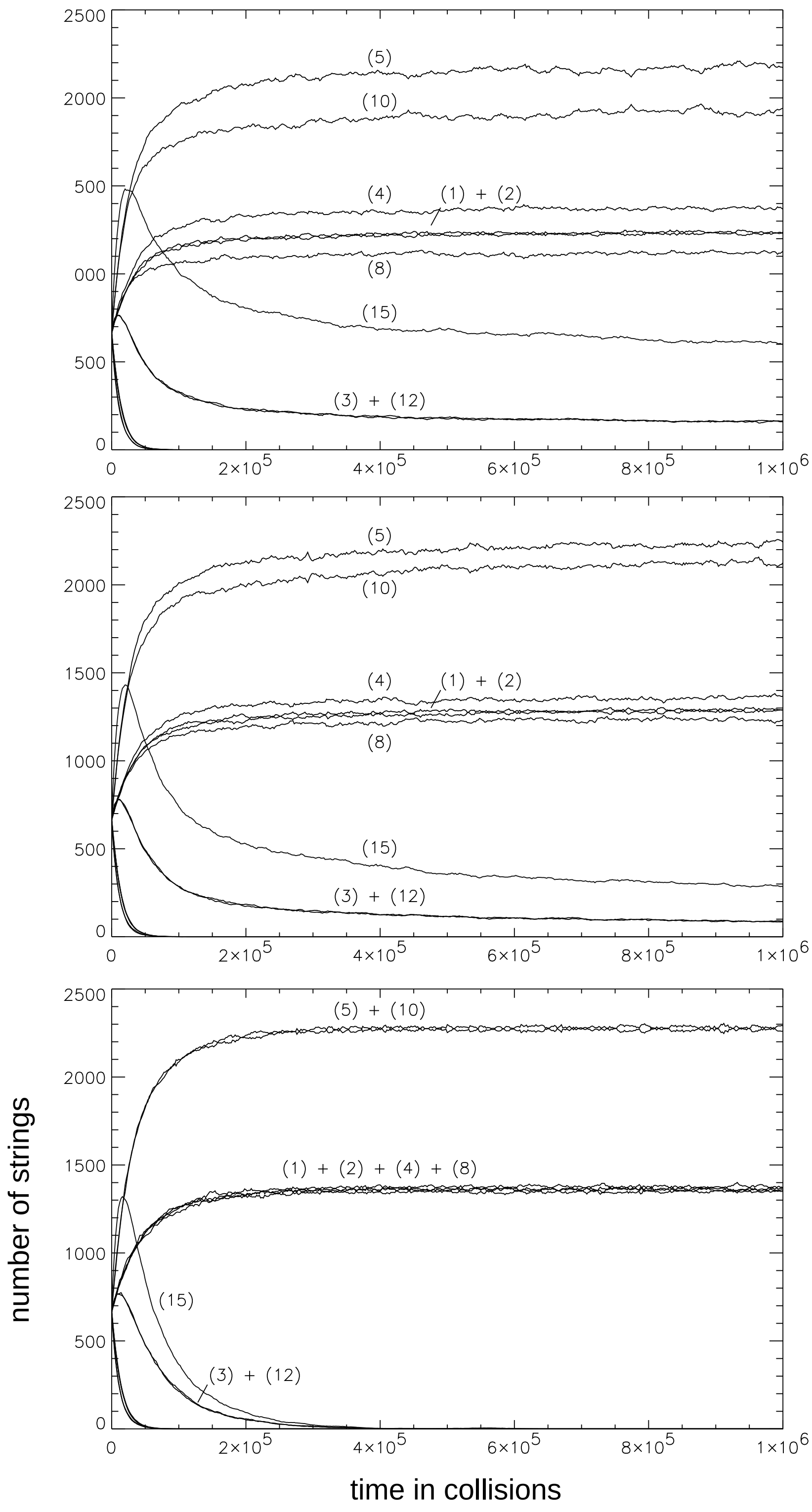

Fig. 3. Behavior of three different systems with Moore neighborhood (top); Gaussian neighborhood (middle); and with infinite neighborhood, i.e. without topology (bottom). Initial state: homogeneous distribution of strings, folding: topological. Concentrations of string types have been averaged over 10 independent runs for 10^6 reactions. (See text.)

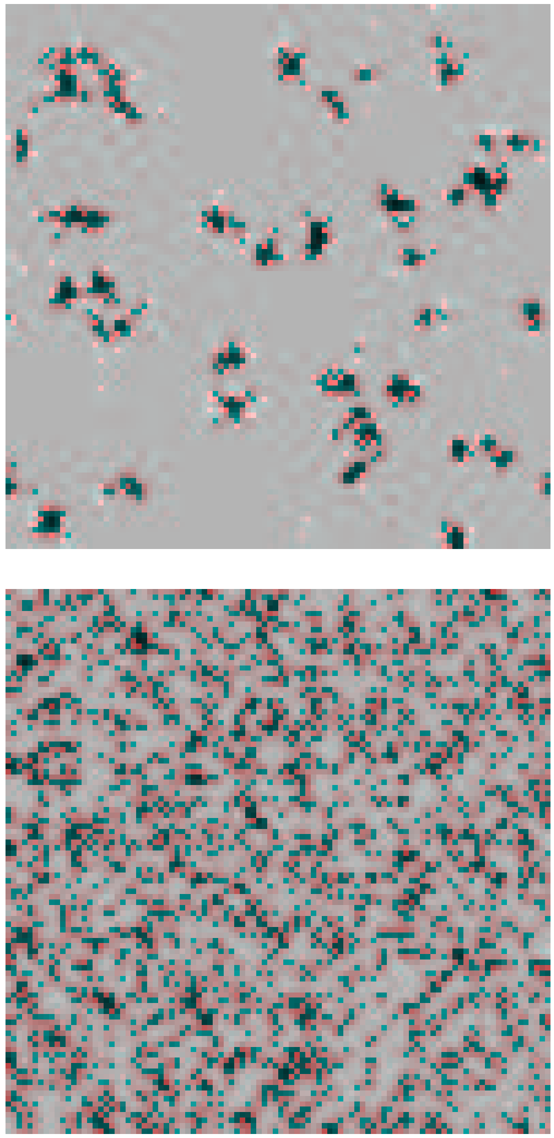

Fig. 4. Top: Spatial distribution of $s^{(15)}$ strings in clusters ($s^{(15)}$ = black, $s^{(3)}$ and $s^{(12)}$ = white, others = gray). Bottom: Uniform distribution of $s^{(5)}$-strings ($s^{(5)}$ = black, others = gray). Typical outcome after $2 \cdot 10^7$ reactions. System configuration: homogeneous initial state, topological folding, Moore neighborhood.

that $s^{(15)}$ has no chance to survive in the infinite neighborhood case whereas in the former two cases it approaches some finite equilibrium levels (after $2 \cdot 10^7$ reactions, not shown here).

The reason for this behavior can be found in the tendency of $s^{(15)}$ to form clusters. Figure 4 contrasts the behavior of string type $s^{(15)}$ (top) with that of $s^{(5)}$. There is a clear preference for clusters on the part of $s^{(15)}$.

For a formal definition of clusters we adopt the following notation: Let Z be the set of all points and $D \subset Z \times Z$ a relation on Z. Two points $z, \overline{z} \in Z$

belong to D, $(z, \overline{z}) \in D$, if z and $\overline{z}$ are direct neighbors in the sense of the Moore neighborhood.

Let $C \subseteq Z \times Z$ be another relation on Z. For $z, \overline{z} \in Z$ we can define:

$$(z, \overline{z}) \in C \overset{\text{def}}{\iff} \begin{cases} z = \overline{z} \\ \vee\ (z, \overline{z}) \in D \\ \vee\ \exists \overline{\overline{z}} \in Z : (z, \overline{\overline{z}}) \in C \wedge (\overline{\overline{z}}, \overline{z}) \in C \end{cases}$$

C is an equivalence relation and we can partition Z into equivalence classes of C. These classes are the clusters, and their size is determined by counting the elements of these classes.

In Figure 5 the average size of clusters is depicted, again for the three different types of neighborhood considered earlier. The cluster size of $s^{(15)}$ clearly dominates if it can survive at all.

One could argue that all types of strings, even in a random distribution of strings would come in clusters of different sizes. Figure 6 therefore, compares the distribution of cluster sizes resulting from random placement of strings and the distribution of sizes after the system has developed for $2 \cdot 10^7$ reactions. Whereas there is no difference for many string types, $s^{(15)}$ strongly differs from what ought to be expected from a random distribution.

For comparison, we show in Figure 7 how a homogeneous initial state develops when using the non-topological folding. Here, $s^{(15)}$ is able to survive even without local niches, although the neighborhood clearly has some influence on its success. Again, $s^{(15)}$ forms clusters. The reason for the different behavior between these two variants of the system will become clear in Section 5.

4.2 Inhomogeneous initial state

So far, we have seen that the system can develop from maximal disorder (uniform distribution of string types at the outset) to a state of considerable order. Now we would like to study what happens if we start with an ordered state at the outset. This means, we need an inhomogeneous distribution of species in the beginning.

Because we found out in earlier simulations that only the subset of string types $(s^{(1)}, s^{(2)}, s^{(3)}, s^{(4)}, s^{(5)}, s^{(8)}, s^{(10)}, s^{(12)}, s^{(15)})$ was able to survive in the long run, we conducted some runs with only these sorts, starting from inhomogeneous initial states. A reduced reaction table of this subset is found in Table 3. Note

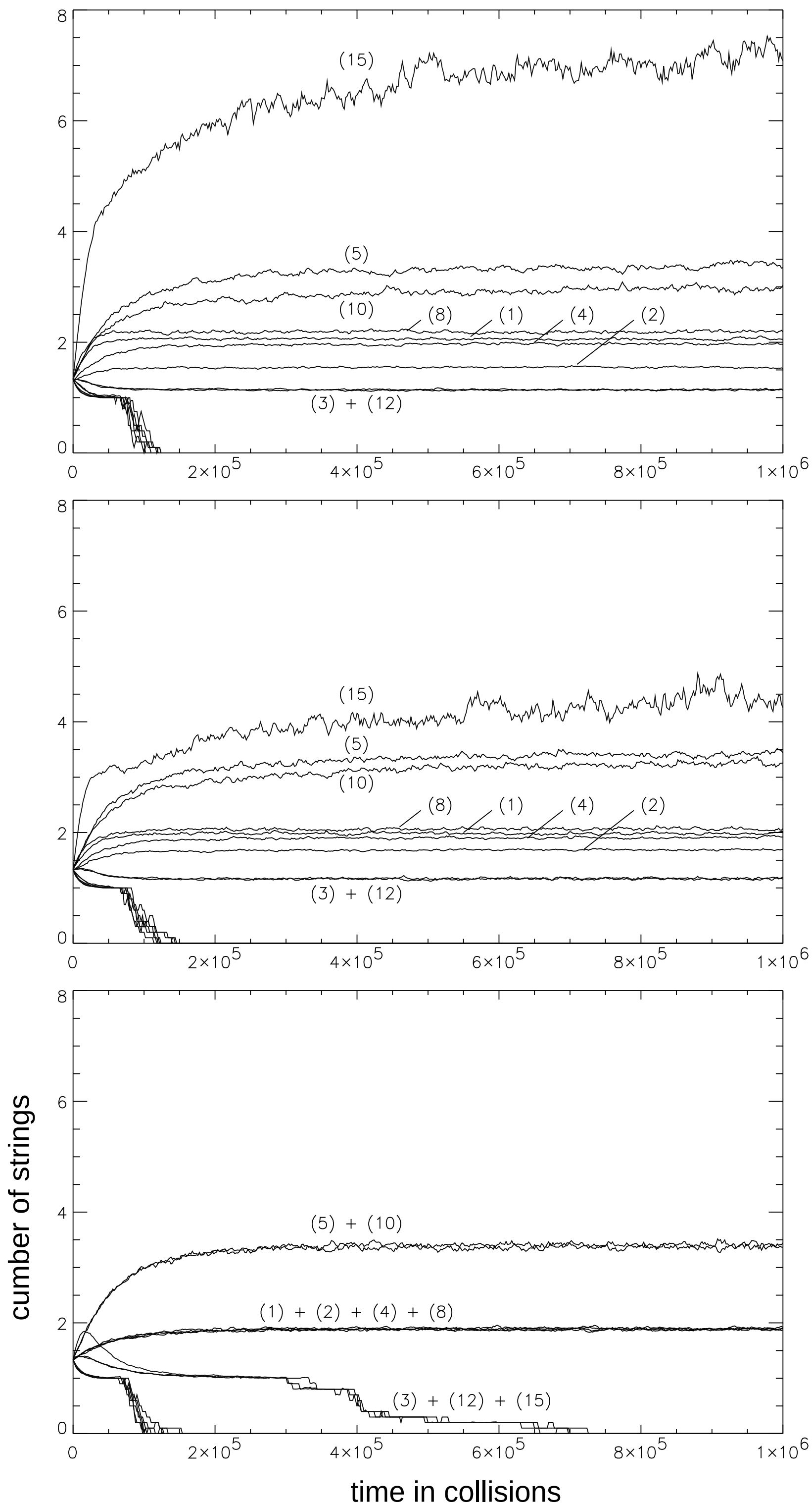

Fig. 5. Average cluster size for different systems during 10^6 reactions (10 independent runs averaged). System configuration: homogeneous initial state, topological folding, Moore neighborhood (top); Gaussian neighborhood (middle); and infinite neighborhood (bottom).

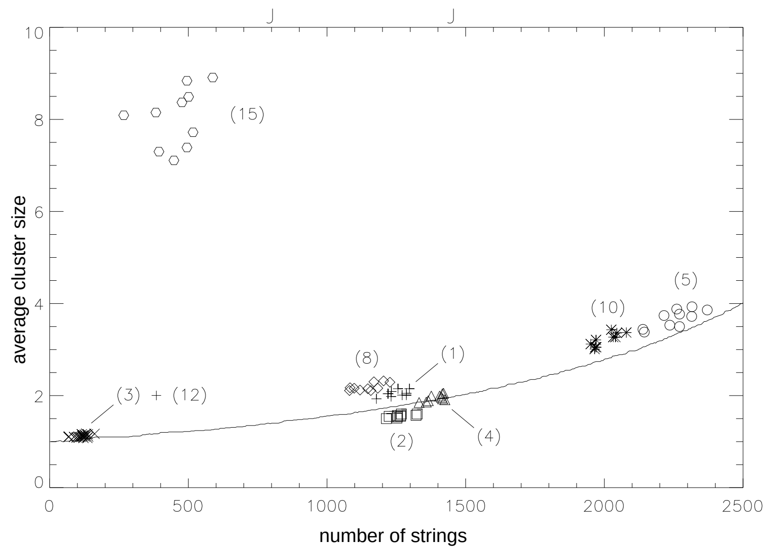

Fig. 6. Average cluster size for string types. Compared is the resulting distribution after a random placement of strings (solid line), with the one after the system has been run 10 times for $2 \cdot 10^7$ reactions (dots). String $s^{(15)}$ is clearly showing a much larger tendency to cluster than what had to be expected. System configuration: homogeneous initial state, topological folding, Moore neighborhood.

Operator	String								
	1	2	3	4	5	8	10	12	15
1	1	0	1	4	5	0	0	4	5
2	0	1	1	0	0	4	5	4	5
3	1	1	1	4	5	4	5	4	5
4	0	2	2	0	0	8	10	8	10
5	1	2	3	4	5	8	10	12	15
8	2	0	2	8	10	0	0	8	10
10	2	1	3	8	10	4	5	12	15
12	2	2	2	8	10	8	10	8	10
15	3	3	3	12	15	12	15	12	15

Table 3
Reduced reaction table (topological folding)

that we can indeed reduce the system to these reactions, if only those types are present in the beginning. As we mentioned earlier, reactions producing $s^{(0)}$ are prohibited anyway.

Figure 9 depicts the development of systems with four different inhomogeneous

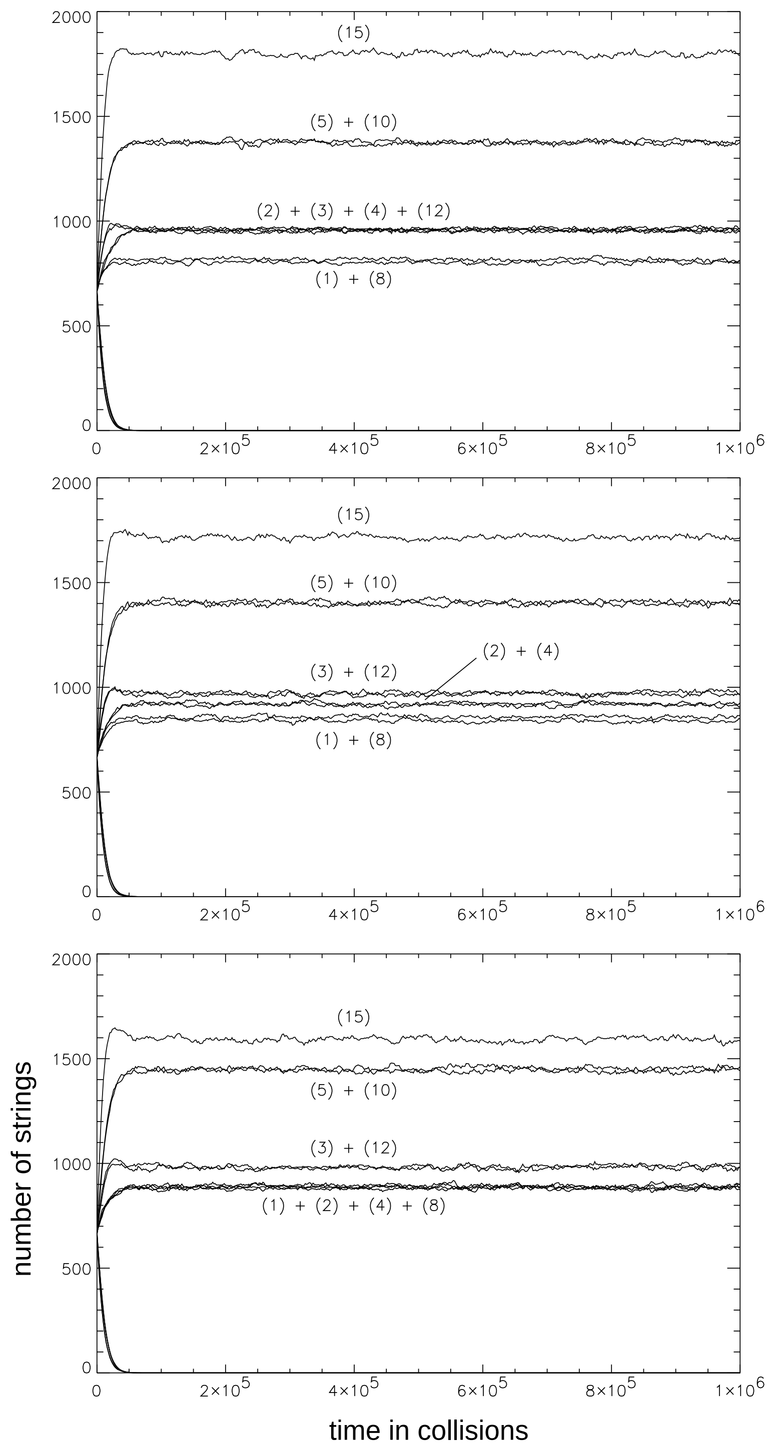

Fig. 7. Behavior of three different systems with Moore neighborhood (top); Gaussian neighborhood (middle); and with infinite neighborhood, i.e. without topology (bottom). Initial state: homogeneous distribution of strings, folding: non-topological. Concentrations of string types have been averaged over 10 independent runs for 10^6 reactions.

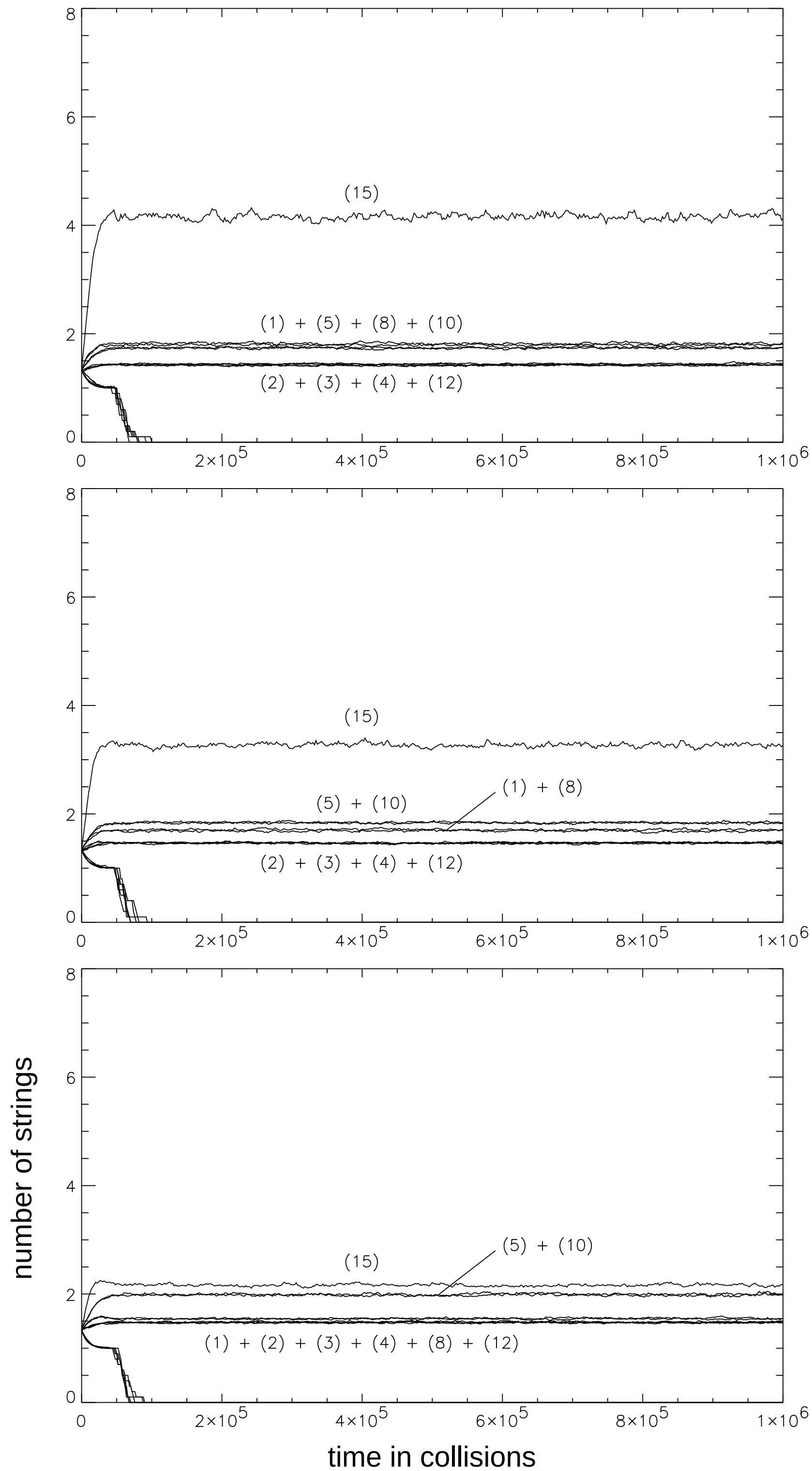

Fig. 8. Average cluster size for different systems during 10^6 reactions (10 independent runs averaged). System configuration: homogeneous initial state, non-topological folding, Moore neighborhood (top); Gaussian neighborhood (middle); and infinite neighborhood (bottom).

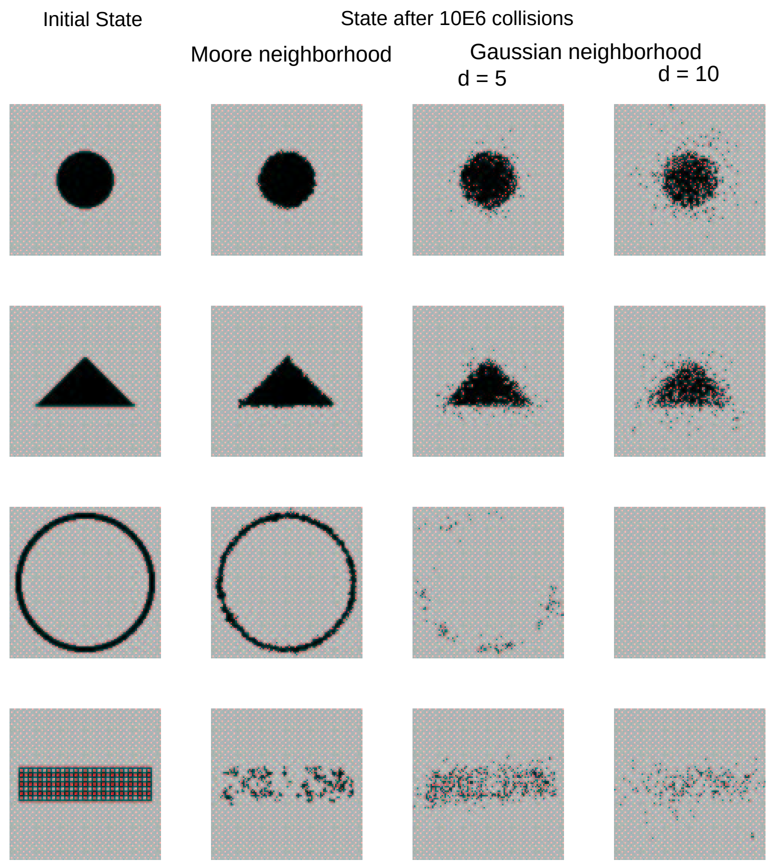

Fig. 9. Different inhomogeneous distributions of $s^{(15)}$ (first row) in the initial state develop differently. Other rows: Status after 10^6 iterations (nearly no changes happen afterwards). The situation depends on the neighborhood interaction. ($s^{(15)}$ = black, others = gray)

distributions of $s^{(15)}$. We simply distributed $\frac{1}{9}$ of the population (the share of $s^{(15)}$) very purposefully. We can observe that this cluster is all the more stable the more compact the initial state was. This has to do with the neighborhood interactions that partially lead to a disappearance of $s^{(15)}$.

With one exception, the structures stay mainly as they are, even after ten times as many iterations have elapsed.

Notably, a large size of the Gaussian neighborhood does not lead to a dissolution of the circular cluster. Figure 10 shows more details of the resulting dynamical pattern which is conserved up to fluctutations.

$s^{(15)}$-strings at the border of the cluster generate $s^{(3)}$, $s^{(12)}$ and $s^{(15)}$ strings. The $s^{(15)}$ strings outside quickly disappear again. Sorts $s^{(3)}$ und $s^{(12)}$ are not able to survive either, since they do not reproduce. The only string types that can intrude the cluster are $s^{(5)}$ and $s^{(10)}$. After having intruded, however, $s^{(15)}$ can "digest" them and replicate itself. $s^{(1)}$, $s^{(2)}$, $s^{(4)}$ and $s^{(8)}$ do not change the $s^{(5)}$- and $s^{(10)}$ string distribution, but feed on $s^{(3)}$, $s^{(12)}$ and $s^{(15)}$. In total, there

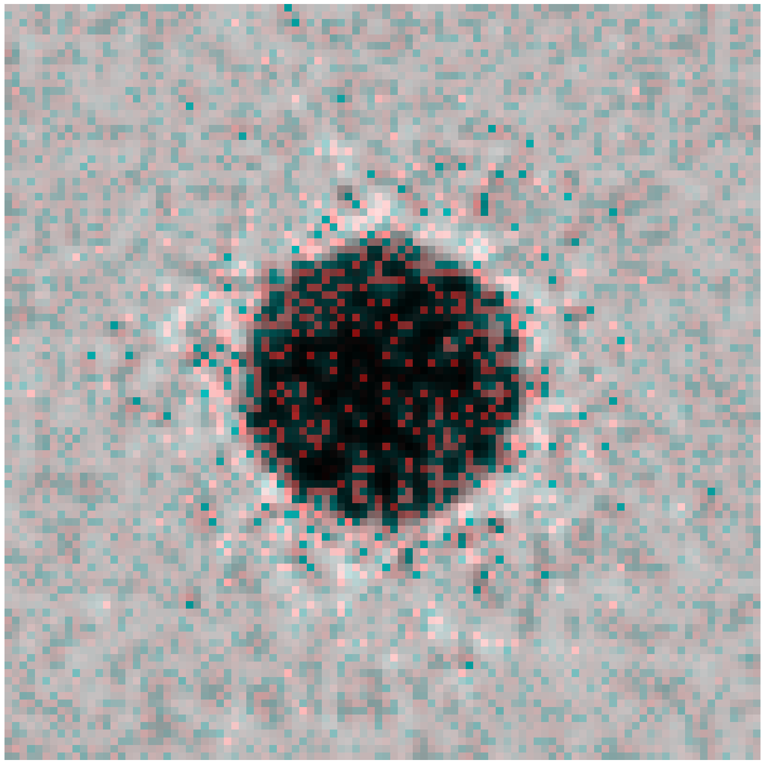

Fig. 10. Magnified details of picture from Figure 9. Typcial state after 10^6 reactions. $s^{(15)}$ = black, $s^{(5)}$ and $s^{(10)}$ = dark gray, $s^{(3)}$ and $s^{(12)}$ = white, others = gray). Inhomogeneous initial state (circular $s^{(15)}$-cluster), Gaussian neighborhood with $d = 10$.

is an equilibrium of production and destruction of string types stabilizing the system.

One can therefore identify four different subsets of strings:

(i) $s^{(15)}$
(ii) $s^{(5)}$ and $s^{(10)}$
(iii) $s^{(1)}$, $s^{(2)}$, $s^{(4)}$ and $s^{(8)}$
(iv) $s^{(3)}$ and $s^{(12)}$

We shall now distribute those different types in specific areas. All areas are allowed to border all others, so that we can better observe the interaction patterns. Figure 11 shows a typical example.

Within the area of group 4 (white) only strings of group 3 (gray) are produced. $s^{(3)}$ and $s^{(12)}$ are consumed by group 3 strings and disappear. $s^{(15)}$-strings (black) can venture into the area of group 2 (dark gray) and take over this area as well. At the end, we have only two separate areas left, one occupied from $s^{(15)}$ the other from group 3 strings. There is a membrane-like boundary layer between the two areas that inhibits changes to propagate from one area to the other. It can be concluded that the initial distribution of strings exerts

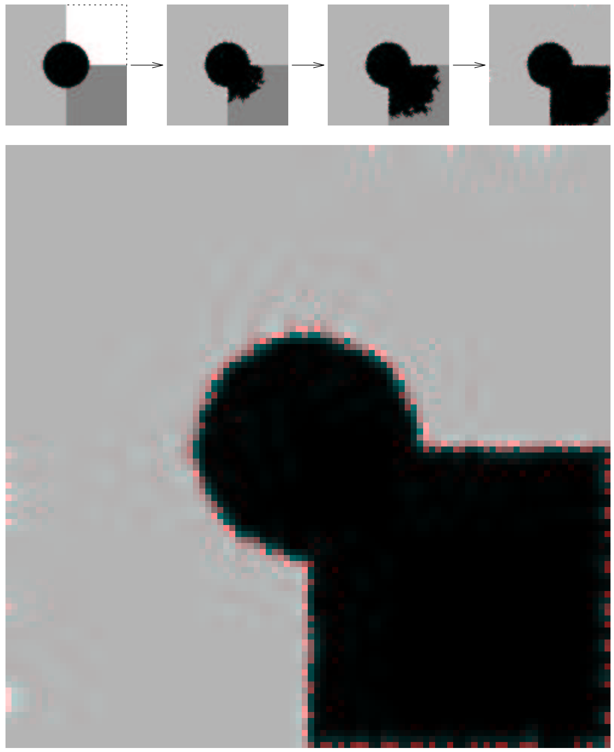

Fig. 11. Typical example of state after 440 000 reactions (small pictures: Intermediate states (110 000 reactions each). $s^{(15)}$ = black, $s^{(5)}$ and $s^{(10)}$ = dark gray, $s^{(3)}$ and $s^{(12)}$ = white, others = gray). Moore neighborhood.

a strong influence on the development of the system.

Finally we want to study the proliferation of $s^{(15)}$ under the influence of different neighborhoods. Initially let the grid only contain $s^{(5)}$ strings and one $s^{(15)}$ string in the center. This is an unstable state which will ultimately develop into a state with $s^{(15)}$ dominating and $s^{(5)}$ extinct. In Figure 12 we show the take-over time (in terms of iterations) for different neighborhoods. ¿From this figure we can clearly see, that the original system is a special case of the topological organizations we studied here.

5 Analysis of reaction tables

In trying to understand the behavior of the system, one might first look at the number of reactions that produce certain strings. Figure 13 shows these numbers. Note that this is independent of the type of folding that is applied, since folding ultimately means a renumbering of string types only.

If a string has a low value in the above figure, it usually leads to extinction. A high value, on the other hand, does not guarantee survival. If we measure

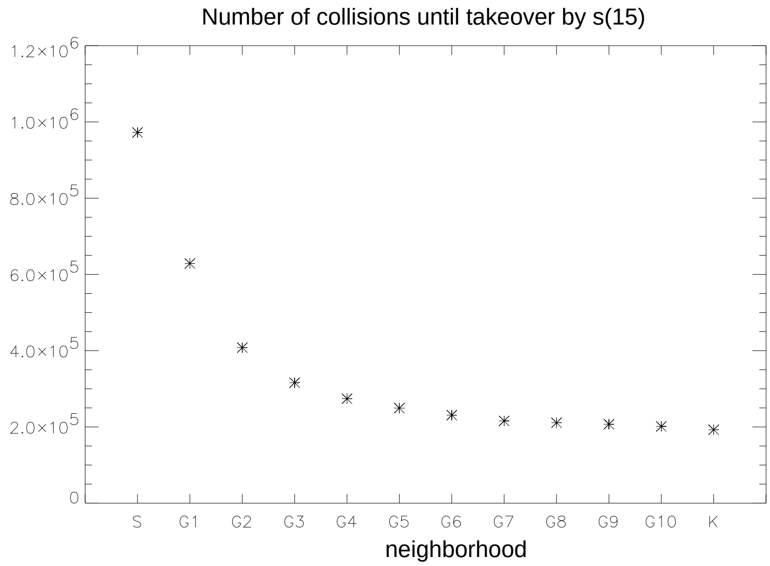

Fig. 12. Number of iterations until takeover by $s^{(15)}$, depending on neighborhood. (averages from 10 independent runs); S = Moore neighborhood, Gn = Gaussian neighborhood with width n, K = no (infinite) neighborhood (well-stirred case)). System configuration: inhomogeneous initial state (one $s^{(15)}$ string, $s^{(5)}$ otherwise).

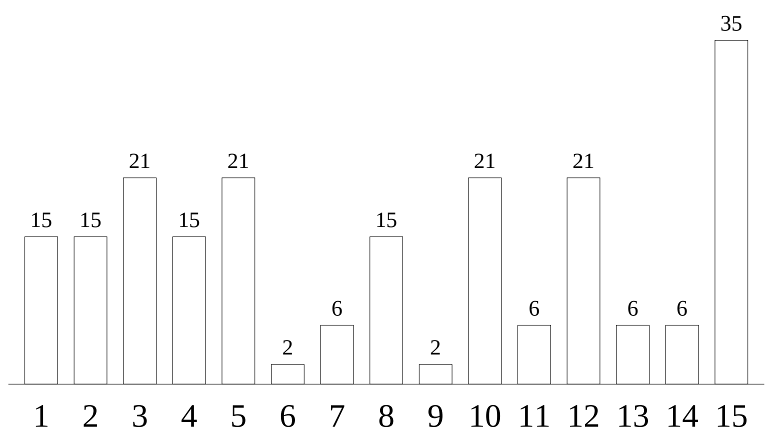

Fig. 13. Number of reactions that produce a certain string type. Of the 256 different possible reactions 49 are prohibited, the rest distributes among types $s^{(1)}...s^{(15)}$. Identical for all foldings.

the number of self-replicating string types in both the topological and non-topological folding, we can see that it is exactly the same (five).

So what does make the difference in behaviour between the two different folding types? A much clearer picture emerges if we look in more detail, how often a string is produced in a replication reaction. Due to the specific reactions we have to discern between the following two possibilities for replication (exluding self-replication):

Topological folding

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

Non-topological Folding

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

Table 4
Degree of duplication (DOD) and degree of conservation (DOC) for all string types in both, the topoligical and the non-topological folding.

(i) Operator and result are identical
(ii) Operand and result are identical

The former we shall call degree of duplication ("DOD"), the latter degree of conservation ("DOC"). More formally,

$$\mathrm{DOD}(s) = k \overset{\text{def}}{\Longleftrightarrow} \begin{cases} \text{there are different } s_1, \ldots, s_k \in \mathbb{B}^N \text{ with} \\ \mathcal{F}(s) \otimes s_i = s \ \wedge \ s_i \neq s \ \ (\text{for } i = 1, \ldots, k) \ \text{ and } \ k = \max. \end{cases}$$

$$\mathrm{DOC}(s) = k \overset{\text{def}}{\Longleftrightarrow} \begin{cases} \text{here are different } s_1, \ldots, s_k \in \mathbb{B}^n \text{ with} \\ \mathcal{F}(s_i) \otimes s = s \ \wedge \ s_i \neq s \ \ (\text{for } i = 1, \ldots, k) \ \text{ and } \ k = \max. \end{cases}$$

Table 4 shows both values for all string types for both foldings.

There obiously is a progressive and a conservative force necessary for a string to be able to survive. The progressive force will allow a string to venture into other areas (not yet occupied by it), the conservative force will allow it to consolidate a given terrain, once occupied. $s^{(15)}$ has both, a high DOD and high DOC, allowing this species to spread in certain areas that have the right "food" for it to thrive, and then subsequently to defend the area occupied.

So why does this species not spread all over the grid? This has to do with the decrease in food supply. As soon as only $s^{(5)}$ and $s^{(10)}$ are left on which to feed, $s^{(15)}$ growth levels out, because both species are able to produced themselves though other species by consumption of $s^{(15)}$ (see Table 3).

5.1 Metabolisms

A **metabolism** Met is a subset of $\mathbb{B}^N$ fulfilling the following conditions:

(i) Met does not contain $s^{(0)}$.
(ii) Met is closed in that all string sorts in Met only produce string types also in Met.
(iii) Each type $s \in Met$ is produced by at least one other member of Met acting as an operator on a string $s_2 \in Met$ with $s \neq s_2$.

More formally this reads:

$$Met \text{ is a metabolism}$$

$$\overset{\text{def}}{\Longleftrightarrow} \begin{cases} s^{(0)} \notin Met \\ \wedge\ (\forall s_1, s_2 \in Met:\ \mathcal{F}(s_1) \otimes s_2 \in Met \cup \{s^{(0)}\}\ \wedge\ \mathcal{F}(s_2) \otimes s_1 \in Met \cup \{s^{(0)}\}) \\ \wedge\ (\forall s \in Met\ \ \exists s_1, s_2 \in Met:\ \mathcal{F}(s_1) \otimes s_2 = s\ \wedge\ s \neq s_2) \end{cases}$$

Figures 14 and 15 depict all the metabolisms of both foldings, ordered hierarchically. The couplings in this metabolism graph are very important. When starting from a homogeneous initial state with topological folding, ususally one of the following metabolisms wins out:

(i) $\{s^{(1)}, s^{(2)}, s^{(3)}, s^{(4)}, s^{(5)}, s^{(8)}, s^{(10)}, s^{(12)}, s^{(15)}\}$
(ii) $\{s^{(1)}, s^{(2)}, s^{(4)}, s^{(5)}, s^{(8)}, s^{(10)}\}$

The weaker the connections are in this graph, the more probable is domination by one string alone (see Figure 15).

With non-topological folding, only the first metabolism appeared. It seems to be a curious fact that in both figures that particular metabolism exhibits most connections to other metabolisms.

6 Discussion and conclusions

Three survival strategies have been observed that have had at least some degree of success:

(i) Cooperation of string species;
(ii) Use of spatial niches;
(iii) Parasitic behavior.

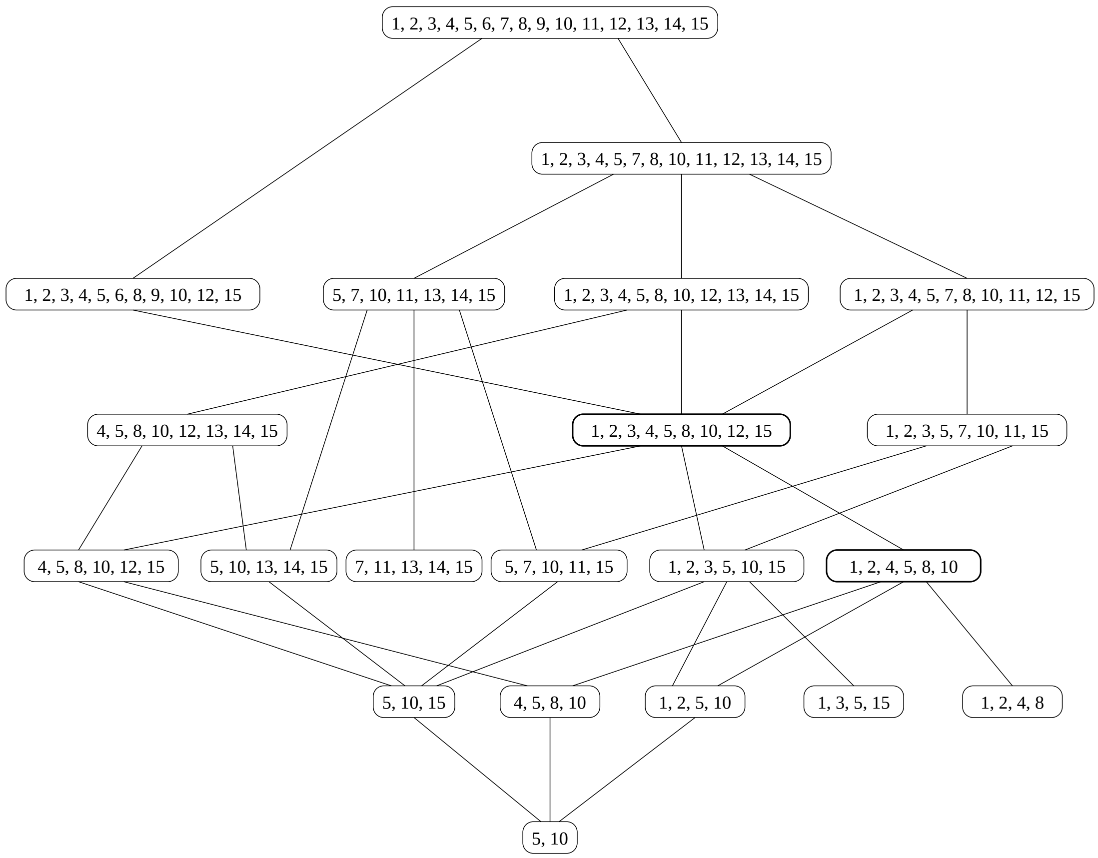

Fig. 14. Hierarchy of metabolisms of topological folding.

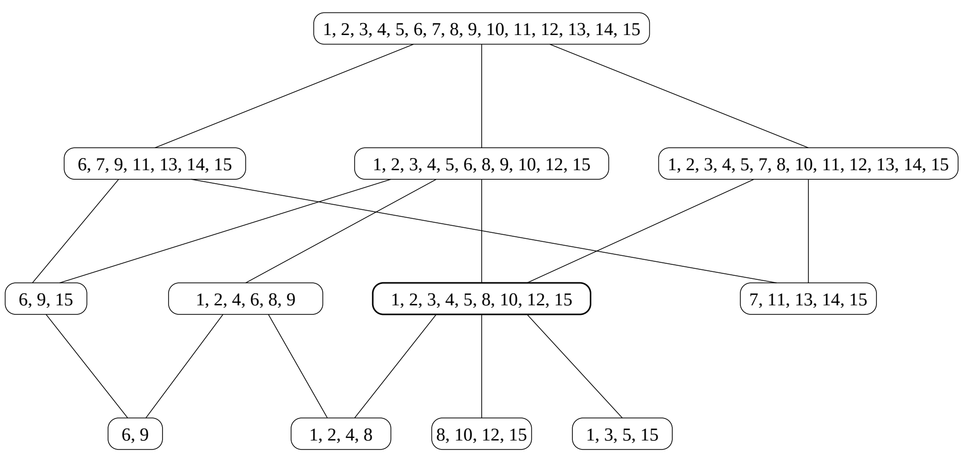

Fig. 15. Hierarchy of metabolisms of non-topological folding.

The metabolisms $\{s^{(5)}, s^{(10)}\}$ and $\{s^{(1)}, s^{(2)}, s^{(4)}, s^{(8)}\}$ are examples of the first strategy. Irrespective of any neighborhood relation, they support themselves and have a certain probability to survive. $s^{(15)}$, on the other hand, has a completely different way of surviving: It can exploit other species for its own benefit. Provided the interaction range of strings is sufficiently restricted, $s^{(15)}$

can hold on to the occupied territories. As a side effect, we observe types $s^{(3)}, s^{(12)}$, living from the existence of $s^{(15)}$.

We can conclude that

– Spatial niches allow higher diversity.
– More spatial interaction results in more homogenity.
– Cooperative interactions are stabilizing species.

A few remarks are in order to compare this system to other ALife systems:

– In contrast to cellular automata [21,22] which also use a grid of points there is no deterministic update rule, as well as no synchroneous updating of elements.
– The difference to Lugowski's computational metabolism [23] is that the abstract maschines he employs are not engaging in a competition.
– Similar to the autocatalytic metabolisms of Bagley und Farmer [24,25] individuals can survive through cooperation.
– As in Fontana's Algorithmic Chemistry [26], functions and arguments are used interchangably. In his case, however, the system is incorporated as the λ-calculus.
– As in Avida, the 2D-variant of Ray's Tierra system [27,28], diversity is increased considerably by introducing topology. But in Avida, spatial stable patterns have not been observed.
– The replicator model of Nuño et al. [29] shows similar spatial clusters as the binary string system studied here.

In the future we would like to study the issue of metabolisms in much more detail, preferably in a more complex system, like with $N = 9$. Also, it will be necessary to develop much more accurate observational tools to study the questions of pattern formation coming with such a system.

ACKNOWLEDGMENT

Support has been provided by the DFG (Deutsche Forschungsgemeinschaft), under grant Ba 1042/2-1 and Ba 1042/2-2. W.B. acknowledges partial support by the International Computer Science Institute, UC Berkeley, CA.

References

[1] H. Haken. *Synergetics – An Introduction.* Springer, Berlin, 2. edition, 1983.

[2] S. Kauffman. *The Origins of Order : Self-Organization and Selection in Evolution.* Oxford University Press, Oxford, 1991.

[3] B.-O. Kueppers. *Information and the Origin of Life.* MIT Press, Cambridge, MA, 1990.

[4] C. G. Langton, editor. *Artificial life: the proceedings of an interdisciplinary workshop on the synthesis and simulation of living systems held September, 1987 in Los Alamos, New Mexico*, volume 6 of *SFI studies in the sciences of complexity*, Redwood City, CA, 1989. Addison-Wesley.

[5] C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, editors. *Artificial life II: proceedings of the workshop on artificial life held February, 1990 in Santa Fe, New Mexico*, volume 10 of *SFI studies in the sciences of complexity*, Redwood City, CA, 1992. Addison-Wesley.

[6] C. G. Langton, editor. *Artificial life III: proceedings of the workshop on artificial life held June, 1992 in Santa Fe, New Mexico*, volume 17 of *SFI studies in the sciences of complexity*, Reading, MA, 1994. Addison-Wesley.

[7] R. A. Brooks and P. Maes, editors. *Artificial life IV: proceedings of the fourth international workshop on the synthesis and simulation of living systems*, Cambridge, MA, 1994. The MIT Press.

[8] C. G. Langton and T. Shimohara, editors. *Artificial life V: Proceedings of the Fifth Workshop on Artificial Life*, Cambridge, MA, 1997. MIT Press.

[9] W. Banzhaf. Self-replicating sequences of binary numbers. *Computers and Mathematics with Applications*, 26(7):1–8, 1993.

[10] M.W. Gray and R. Cedergren. The new age of rna. *FASEB*, 7:4, 1993.

[11] M. Eigen, R. Winkler-Oswatitsch, and P. Woolley. *Steps towards Life: A prespective on Evolution.* Oxford University Press, Oxford, 1992.

[12] K. Kruger, P. J. Grabowski, A. J. Zaug, J. Sands, D. E. Gottschling, and T. R. Cech. Self-splicing rna: autoexcision and autocyclization of the ribosomal rna intervening sequence of tetrahymena. *Cell*, 31:147–157, 1982.

[13] C. Guerrier-Takada, K. Gardiner, T. Marsh, N. Pace, and S. Altman. The rna moiety of ribonuclease p is the catalytic subunit of the enzyme. *Cell*, 35:849–857, 1983.

[14] G. Jetschke. *Mathematik der Selbstorganisation: qualitative Theorie nichtlinearer dynamischer Systeme und gleichgewichtsferner Strukturen in Physik, Chemie und Biologie.* Vieweg, Braunschweig, 1989.

[15] W. Banzhaf. Self-replicating sequences of binary numbers. foundations i: General. *Biological Cybernetics*, 69:269–274, 1993.

[16] W. Banzhaf. Self-replicating sequences of binary numbers. foundations ii: Strings of length n=4. *Biological Cybernetics*, 69:275–281, 1993.

[17] W. Banzhaf. Self-organization in a system of binary strings. In Brooks and Maes [7], pages 109–118.

[18] W. Banzhaf. Self-organizing algorithms derived from rna interactions. In W. Banzhaf and F. Eeckman, editors, *Evolution and Biocomputation*, pages 69–102, Berlin, 1995. Springer.

[19] W. Banzhaf. Self-replicating sequences of binary numbers — the build-up of complexity. *Complex Systems*, 8:205–215, 1994.

[20] Peter F. Stadler, Walter Fontana, and John H. Miller. Random catalytic reaction networks. *Physica D*, 63:378–392, 1993.

[21] J. v. Neumann. *Theory of self-reproducing automata (edited and completed by A. W. Burks)*. University of Illinois Press, Urbana, IL, 1966.

[22] E. R. Berlekamp, J. H. Conway, and R. K. Guy. *Gewinnen: Strategien f"ur mathematische Spiele (Band 4: Solitairspiele)*. Vieweg, Braunschweig, 1985.

[23] M. W. Lugowski. Computational metabolism: towards biological geometries for computing. In Langton [4], pages 341–368.

[24] R. J. Bagley and J. D. Farmer. Spontaneous emergence of a metabolism. In Langton et al. [5], pages 93–140.

[25] R. J. Bagley, J. D. Farmer, and W. Fontana. Evolution of a metabolism. In Langton et al. [5], pages 141–158.

[26] W. Fontana. Algorithmic chemistry. In Langton et al. [5], pages 159–209.

[27] C. Adami and C. T. Brown. Evolutionary learning in the 2d artificial life system "avida". In Brooks and Maes [7], pages 377–381.

[28] T. S. Ray. An approach to the synthesis of life. In Langton et al. [5], pages 371–408.

[29] J. C. Nuno, P. Chacón, A. Moreno, and F. Morán. Compartimentation in replicator models. In F. Morán, A. Moreno, J. J. Merelo, and P. Chacón, editors, *Advances in artificial life: third european conference on artificial life, Gra*, pages 116–127, Berlin, 1995. Springer.