

# A search for multiple autocatalytic sets in artificial chemistries based on boolean networks

Harald Hüning

Electrical Engineering, Imperial College, London SW7 2BT, UK

## Abstract

Populations of strings which interact in ways defined by an artificial chemistry can self-organise spontaneously into an autocatalytic set. This paper considers populations of binary strings with fixed length and a reaction scheme that uses strings as both data (or tape) and machine. Here the machine is a boolean network where some parameters are determined by a string from the population. The input to the machine is given by a second string drawn from the population. In the artificial chemistry based on boolean networks, simulations have revealed a high sensitivity on a probabilistic rate that filters out trivial patterns. By variation of the rate parameter, multiple stable sets have been found. Short string lengths are used here, in order not to rely only on simulations, but also to keep the reaction graph small enough to be able to search for possible autocatalytic sets. A search method has been developed that finds all closed subgraphs of the reaction network, which indicate to a high degree what autocatalytic sets are possible. While simulations most often give only one as a result, the search saves many simulation runs, because it is independent of the initial populations. The resulting number and size of autocatalytic sets gives information about any freedom of the system to adapt, e.g. when coupling such a system to an environment that can impose constraints. So this description of the behaviour of artificial chemistries appears useful for further artificial studies of molecular evolution and the origin of life.

## Introduction

Artificial chemistries make abstractions from the natural chemistry in the representation of molecules, populations, and the implementation of reactions. Although artificial, they have already provided simulations of the cooperation of molecules in metabolic or catalytic networks (Farmer, Kauffman & Packard, 1986), and they may also serve to discover laws of evolution (Bagley, Farmer & Fontana, 1992).

The study of artificial chemistries has the advantage that in computer simulations all details can be made observable, and more than one trajectory of the dynamics can be investigated. The general questions are

first, when do self-replicating units appear, starting only from the existence of simpler elements, and given some abstract chemical rules? Secondly, how can replicating units evolve, and do any effects broadly share laws with evolution and the origin of life in the chemistry of nature?

Farmer, Kauffman & Packard (1986) have first simulated interacting strings with concatenation and cleavage reactions. They have found an *autocatalytic set*, i.e. a set of strings which are all produced from reactions within the set. Another way of defining the chemistry is based on the notion of using strings as data and machines, where the machines are automata that transform strings. Ikegami & Hashimoto (1995) have used different types of strings for data and machine, while Dittrich and Banzhaf (1998) have used just one type of string for both data and machine. Replication arises from closed subgraphs of the random catalytic chemistry (Stadler, Fontana & Miller, 1993), where data and machine correspond to two molecules catalysing a reaction.

Regarding the evolution of autocatalytic sets, Bagley, Farmer & Fontana (1992) have studied evolutionary modifications of reaction graphs, and have found that autocatalytic sets can change with varying degrees. In the data-machine type of simulation, Dittrich and Banzhaf (1998) have observed an exchange of string segments in one example. This is reminiscent of a mutation operator, but it acts on individual strings rather than on replicating units like autocatalytic sets.

All these studies suggest that autocatalytic sets typically involve only a small part of the reaction network, and in the large networks defined by artificial chemistries no other method than simulations of the dynamics (or fixed-point solutions) seems to give information about these sets.

## Motivation

The current work addresses the question of how many different autocatalytic sets are possible in a given ar-

*Copyrighted Material*

tificial chemistry. Simulations give only one set as a stable result, and sometimes more sets can be recognised by their competition. When there would be an abundance of possible sets in a system, then one could regard the particular strings in a set as somewhat arbitrary, or having degrees of freedom. This is important for further studies of interactions with an environment, e.g. imposing constraints on the survival of strings or on selected reactions, or injecting strings into the population considered. If there is a degree of freedom in the strings that can exist in a dynamically stable population, no matter in which autocatalytic set, then such interactions with the environment can lead to decisions that affect the convergence of the population. In contrast, if there would be always only one autocatalytic set for one artificial chemistry, then there is not much freedom for decisions. However, as Bagley, Farmer & Fontana (1992) have studied, the coming into existence of a single molecule can change an autocatalytic set dramatically.

Most examples of autocatalytic sets in the literature are lacking a characterisation of the number and size of the sets. The sets reported for example by Farmer, Kauffman & Packard (1986) and Bagley, Farmer & Fontana (1992) are very large, and it is very hard to gain further insight into the dynamics. So it seems desirable to study smaller systems, where for example the difference between the connected graphs of reactions and the typically smaller autocatalytic sets can be investigated.

Here smaller string lengths are chosen to reduce the size of the reaction graph. The system uses strings as both data and machine (Dittrich & Banzhaf, 1998), because this type of chemistry admits in principle the construction of arbitrary new strings. Dittrich and Banzhaf (1998) mention a phase where the the system explores the space of strings. However, the automaton (like a computer processor) that they have used, does not seem appropriate for use with much shorter strings, since bit-words are translated into machine instructions.

A different scheme for transforming bit-strings is given by boolean networks (Kauffman, 1969), sometimes also called weightless neural networks (Aleksander & Morton, 1990). The artificial chemistry is here defined by means of boolean networks for the following reasons. Boolean networks allow relatively general functions depending on the number of bits that each node reacts to. They implement discrete functions like Dittrich and Banzhaf's (1998) automaton, so it is expected that the system has autocatalytic sets with similar characteristics. Most importantly, small string lengths can be chosen in a systematic way, since

the strings need not be translated into a series of instructions.

Thus the particular questions addressed in this paper are:

1. What is the number and size of the autocatalytic sets?
2. How do autocatalytic sets relate to properties (subgraphs) of the reaction network?
3. Do the boolean networks have similar characteristics as Dittrich and Banzhaf's (1998) automaton, even with small string lengths?

The behaviour of artificial chemistries has been mainly investigated by simulations and not analytically, due to the large number of different strings involved. For small reaction networks the dynamics can often be analysed deterministically (Eigen & Schuster, 1978). For very large reaction networks stochastic simulations are used, unless a restricted reaction network can be extracted (Bagley, Farmer & Fontana, 1992). The disadvantage of simulations is their dependence on the initial populations, so usually many simulation runs have to be carried out.

This paper aims at characterising the behaviour of artificial chemistries independently of the initial populations. By searching for potentially stable sets in the reaction graph, behaviours may be found that are missed by simulation studies alone. The search can be made quite exhaustively in the simple artificial chemistry with short binary strings, and the applicability to longer strings is discussed below.

Boolean networks have much been studied with the output feeding back directly on the input. The global behaviour of boolean networks has been illustrated as basins of attraction fields (Wuensche, 1997). These are diagrams with a node for every distinct state, and a line for every possible state transition. For any initial state, one can follow the states (trajectory) in the diagram, until one encounters a state again. So the attractors are either a state with a transition to itself, or a state cycle. Kauffman (1969) has studied the cycle lengths of these attractors.

The search in a reaction graph of an artificial chemistry is comparable to, but more complex than searching for state cycles, since boolean networks on their own have deterministic behaviour, but the simulated reactions are stochastic and involve the collisions of two strings.

Dittrich and Banzhaf (1998) have used the following reactor algorithm that carries out the stochastic collisions:

1. Randomly select two strings from a population

- If the reaction  $s_1 + s_2 \Rightarrow s_3$  exists according to a filter, replace a randomly chosen string from the population with  $s_3$

Their automaton uses one string as input or data string, and the other one as encoded machine instructions. The instructions are carried out to process the input string sequentially. For each bit in the input string one bit is written into the output string, so all strings have the same length. A filter condition is applied to avoid self-replication of strings, and the all-0 string is disallowed as well. The results are characterised by an exploration phase with high diversity of the strings in the population, until typically a small number of strings suddenly rises in their population number and takes over the population. For example, an autocatalytic set of eight strings has been reported, where all reactions of any two of the eight strings produce another string of the set. Similar autocatalytic sets are expected using the boolean networks, where shorter string lengths allow searching in the reaction graphs.

The next section introduces the artificial chemistry with boolean networks. In simulations the filter condition is varied. Then follows a graph-based global search for the possible stable sets, which is more efficient than the simulations.

### Artificial chemistry based on boolean networks

The simulation system consists of a population of binary strings and a machine that transforms strings, implementing a reaction scheme (Fig. 1). A boolean network is used as the machine, or in other words to implement a reaction scheme for the population of binary strings.

Fig. 1 illustrates the reactor algorithm mentioned in the introduction. At every time step two strings are chosen randomly (with replacing) from the population, and used by the machine, in this case a boolean network (Fig. 2). One of the strings acts as input to the machine, the other is used to specify parameters of the machine itself, very much like setting the weights of a standard neural network. So the machine can be different in every time step. Processing of an input string gives an output string which is put back into the population if it passes a filter. If the string is put back, it replaces another randomly chosen string. This is the only operation that changes the population in the current simulations.

The filter in Fig. 1 can prohibit certain strings from being put back. Here, the filter is chosen as putting back the strings with a probability that depends on some property of the string, e.g. the number of '1's in

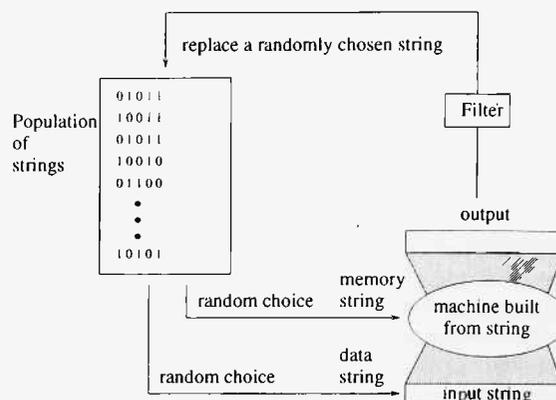


Figure 1: Reactor algorithm with a population and machine. The 'memory string' sets parameters of the machine, which is a boolean network.

it. These probabilities are given directly in a look-up table. Such a filter has similar effects on the dynamics as kinetic rate constants in chemical models.

Details of the machine are shown in Fig. 2. There are as many Random Access Memories (RAMs) as there are bits in the strings. Each RAM has the same number  $n$  of random, but fixed, connections to the input string. The bits conveyed by the connections form a bit-word that addresses a memory location in the RAM. The content of each RAM comes from the memory string as in Fig. 2. In the assignment from the memory string to the  $2^n$  memories in each RAM, every bit is used several times, again in a random but fixed way.

The connections and the string-to-memory assignment are the fixed parameters of the boolean network, while its function changes with the memory string at every time step. The outputs of all RAMs are combined to the output string, and if it is passed through the filter, it is put back into the population.

### Simulation

At the beginning of a simulation, the population is initialised with random strings. Then the reactor algorithm is run. Some examples of reactions from the first simulations are shown in Table 1.

The last three reactions in Table 1 exemplify how the all-0 and all-1 strings are produced from other strings. The all-1 string eventually takes over the whole population, so it 'wins' in this first simulation (see Fig. 3).

When the whole population has converged to the same string, no more changes can happen. The all-0 and all-1 strings are trivial, because they are repro-

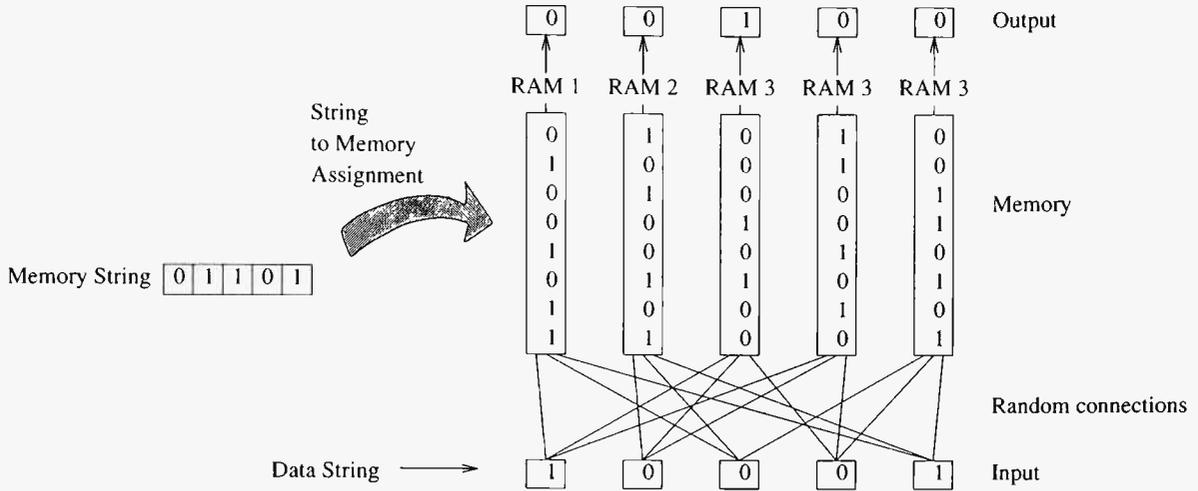


Figure 2: Boolean network that implements the machine with parameters taken from the memory string. For each output bit there is a random access memory (RAM), in this example 5 with  $n = 3$  connections each. Each RAM has  $2^n$  memory locations that are filled with bits from the memory string. Fixed parameters are the random but fixed connections, and the string-to-memory assignment, where each bit is used several times.

$01101 + 10110 \Rightarrow 00010$   
 $10110 + 10011 \Rightarrow 10111$   
 $10111 + 10101 \Rightarrow 10001$   
  
 $10101 + 10110 \Rightarrow 11111$   
 $00000 + 11111 \Rightarrow 00000$   
 $00000 + 00011 \Rightarrow 00000$

Table 1: Reactions from a first simulation. The first string is the memory string, the second is the input. The boolean network has five RAMs with 3 connections each.

duced independently of the connections and memory assignment of the boolean network. How much has the system to be changed in order to stabilise at any other of the  $2^n - 2$  (in this case 30) strings? Other authors have disallowed any self-replication, and the creation of the all-0 string altogether. Here a more moderate filter condition has been found sufficient.

### A first global measurement

The following method shows that the trivial patterns have a much higher chance to be produced from random initial conditions. Globally counting the results of all possible reactions reveals, that for the chosen fixed parameters (connections, memory assignment) there is a much higher number of reactions that produce the trivial strings than others. In particular, all pairwise

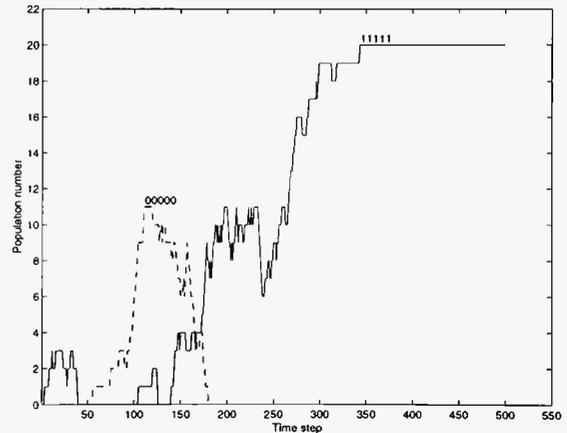


Figure 3: Population of the strings 00000 and 11111 in the first simulation. RAMs and connections as in Table 1. The population size is 20.

combinations of all possible strings are presented to the boolean network, and the occurrences of each of the possible output patterns are counted. Then these are averaged for all strings with the same number of '1's, see Fig. 4.

The histogram shows the average times a string with a given number of '1's is produced over the whole reaction scheme. The higher number of producing all-0

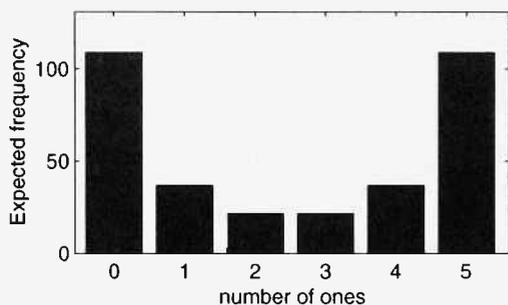


Figure 4: Histogram of the number of '1's, averaged over the  $\binom{5}{k}$  strings with the same number of '1's.

and all-1 strings has been found typical for boolean networks with different random connections and string-to-memory assignments. Assuming an even distribution in the initial string population, the histogram shows the bias towards producing the trivial strings at the beginning of a simulation.

### The filter rate

In contrast to filtering out trivial strings completely, here a probability like a rate or fitness is applied to whether putting back a string into the population or not. For each string that is output from the machine, the number of ones is counted, and the filter probability is looked up in Table 2.

Number of ones	0	1	2	3	4	5
Rate (probability)	$x$	1	1	1	1	$x$

Table 2: Modified filter condition (probability of replacing). For simplicity, only the filter rate for all-0 and all-1 patterns  $x$  is chosen to be variable.

The filter rate  $x$  has been varied in simulations, and surprisingly it is not necessary to fully neutralise the bias in Fig. 4. Merely by changing  $x$  from 1 to 0.95 in the system from Fig. 3 makes the population converge to a different string (11011). For another example of a boolean network (with other connections and string-to-memory assignment), very many changes of the attractor with the variation of the rate  $x$  have been found, see Table 3.

The many different results depending on the rate may be interpreted on the one hand as a lack of robustness of the results, but on the other hand there appear to be several stable results, such that small changes in a parameter can switch between them. There appears to be a potential to add further constraints to the system

rate $x$	stable	steps
0	01110	48700
0.05	10001	1800
0.1	10001	10200
0.15	10001	600
0.2	10001	600
0.25	01110	32200
0.3	10001	2500
0.35	10001	1200
0.4	10001	8700
0.45	10001	11000
0.5	- 0 -	17600
0.55	10001	5300
0.6	01110	1000
0.65	01110	700
0.7	01110	2100
0.75	- 1 -	1300
0.8	- 0 -	600
0.85	- 1 -	1100
0.9	01110	500
0.95	- 1 -	300
1	- 1 -	400

Table 3: Results of simulations where the filter rate  $x$  is varied. Trivial strings are denoted by '- 1 -' or '- 0 -' respectively. The boolean network has five RAMs as before, but now 2 connections each. The population size is 20.

that decide about the attractor. Before the range of possible attractors is investigated, another simulation with more than one resulting string is presented.

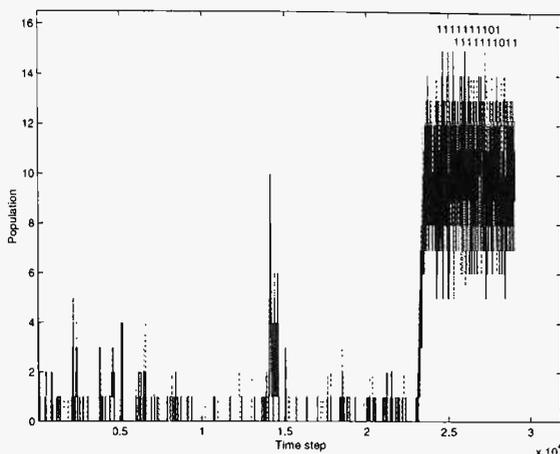


Figure 5: An autocatalytic set. The boolean network has 10 RAMs with 2 connections each. The population size is 20.

### An autocatalytic set

The simulation in Fig. 5 shows a set of strings that co-exist in a stationary way. The population of the two strings keeps fluctuating, but both stay at a high population number, suggesting that they stabilise each other. Fig. 6 shows the reaction pattern underlying this simulation in the stationary state. Clearly none of the strings can reproduce alone. Deterministic equations for this reaction scheme are

$$\begin{aligned} \dot{x}_1 &= x_1 x_2 + x_2 x_2 - x_1 f \\ \dot{x}_2 &= x_1 x_1 + x_1 x_2 - x_2 f \end{aligned} \quad (1)$$

with

$$f = 2x_1 x_2 + x_1^2 + x_2^2$$

Linearisation around the fixed point  $x_1 = x_2 = 0.5$  gives the eigenvalues  $\lambda_1 = -2$  and  $\lambda_2 = -1$ , so the system is locally stable. For more details see Hünig (2000) and Eigen & Schuster (1978).

### A search method

The previous section has shown that simulation results depend on the filter rate  $x$ , and that autocatalytic sets can be found in artificial chemistries based on boolean networks. This section aims at finding the total number of stable sets.

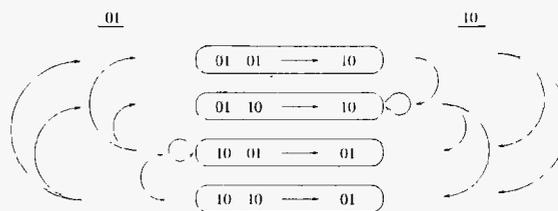


Figure 6: Reactions involved in the autocatalytic set. Nodes are reactions. Arrows show that the two reaction partners can always be produced within the set.

Like attractor cycles can be found by a global search for boolean networks, here cycles are searched for in the reaction graph. However, there are always two strings meeting in a reaction, so the graph is more complex. The reactions requiring the collision of two strings give every single string many possible successors in the graph. So the cycles in the graph can be heavily interconnected, and the dynamics is not confined to a cycle like for recurrent boolean networks.

In a simulation, pairs of strings are chosen randomly, so the reactions have a stochastic order. Over time, the population of strings can be imagined to home in onto a stable autocatalytic set, i.e. a small number of strings closely connected by their reactions.

There are properties of the reaction graph that are indicating the existence of autocatalytic sets:

1. all members of a set are produced from reactions within the set
2. all reactions between the members produce members of the set

The second property may not hold strictly, when dynamic competition reduces the size of the set (see parasites in the discussion). Due to above properties of the reaction graph, possible autocatalytic sets can be searched for, avoiding simulations and their dependence on initial conditions. The search only gives possible sets, because their size can still be reduced by dynamical properties, e.g. competition between subsets. Simulations or analysis of the dynamics can be used after the search to investigate the stability of the sets.

The search for stable sets involves keeping a list of strings as set members, and evaluating all possible reactions between them. All reaction results are added to the list, then the reactions are evaluated again, until no more strings are added. In many cases it may be sufficient to start the list with two strings, but in general more combinations may have to be tried exhaustively. A case where the search has to start from

Set	Type
0	A
0 ... 31	
0, 31	B
14	A
14, 17	B
14, 31	B
17	A
17, 31	B
31	A

Table 4: Results of the search for possible sets. The members of a set are shown as integers, 14 corresponds to 01110, and 17 corresponds to 10001. The second set from the top denotes the whole reaction scheme, while the reaction schemes A and B are given in Table 5. The boolean network is identical to the simulation in Table 3.

Type	Reactions	Stability
A	$31 + 31 \rightarrow 31$	stable
B	$17 + 17 \rightarrow 17$	unstable
	$17 + 14 \rightarrow 17$	
	$14 + 17 \rightarrow 14$	
	$14 + 14 \rightarrow 14$	
C	$17 + 17 \rightarrow 1$	stable
	$17 + 1 \rightarrow 17$	
	$1 + 17 \rightarrow 1$	
	$1 + 1 \rightarrow 17$	

Table 5: Types of reaction patterns and their stability.

more than two strings is for example a singly connected catalytic chain of self-replicating strings.

### Results of the search

With this search method, the sets in Table 4 have been found for the same fixed parameters of the boolean networks as for the simulation in Table 3. Many sets share the same types of reaction graph, so only the type of graph is given in the Table 4, and Table 5 gives the reaction patterns. The search shows that the strings 01110 and 10001 can be expected from simulations, and indeed only these strings, apart from the trivial strings (cf. Table 3).

A similar fixed-point analysis than for equation (1), but for reaction pattern B of Table 5 proves that the pattern of type B is unstable (in contrast to the pattern in Fig. 6). Only one of the strings takes over the whole population.

Set	Type
0	A
0 ... 31	
0, 31	B
1, 17	C
31	A

Table 6: Results of the search for possible sets using an example network with 5 RAMs and 3 connections each.

With different random choices of the connections and string-to-memory assignments of the boolean network, different stable sets are found. There are possible sets with more than two members, where only a subset may be dynamically stable, and more symmetric and unsymmetric reaction patterns for two-member sets.

For example, the sets in Table 6 have been found for one random choice of 3 connections per RAM. Here appears a stable reaction pattern C (see Table 5) of a two-member set, like in Fig. 6 and equation (1).

The search shows that typically there are a number of stable sets, such that small variations in a simulation (like the filter rate) can change the result (which of the sets the simulation converges to). This can be interpreted on the one hand as there being room free to apply constraints to the system. However, the strings that can remain in a stationary population are not arbitrary. The search shows that the total number of different stable sets is rather small. Although one string changes the boolean network (machine) at every step, the whole system has a rather small number of stable sets, corresponding to only a few possible machines made from the strings in a set.

### Discussion

The results of the search may not totally coincide with all sets that can possibly be found by simulation, because:

1. For large sets the dynamics may further reduce the set (e.g. by competition within the closed subgraph)
2. An autocatalytic set may be stable, although it has parasites (see below). Parasitic branches are not detected by the current search.

A parasite is a species that is supported (catalysed) by the set, but does not give back any support (does not have a catalytic link back into the set). The current search method cannot detect parasitic branches, because it adds all reaction results to the set. Parasites cannot simply be detected as dead-end paths,

because here a product exists for every two partners. Filter rates greater than zero do not reduce the graph of reactions. Assuming an autocatalytic set with a parasite, the search algorithm would find a larger set, and it may fail to satisfy the conditions for the set to be closed after all.

In general, even sets with parasitic branches may be dynamically stable, depending on the coupling strengths (Eigen & Schuster, 1978). In this case, detecting parasites would give better results of a search. For parasites that are not only dead-end subgraphs it is still not solved, however, how a graph-based search could be extended to discriminate host and parasite, and possibly to evaluate which of the two the competition is likely to favour. The detection of parasitic branches may become more relevant when some strings are fully filtered out.

What can be expected for longer string lengths, where the exhaustive search even for small starting sets cannot be carried out, because the number of reactions grows with the square of the number of strings? One may still be able to use a sampling version of the search. For longer strings or variable lengths, one may sample the possible strings and follow the reaction graph for a certain depth.

Wuensche (1997) has implemented a similar sampling method for the attractor basins of random boolean networks. This method plots a histogram of the cycle lengths, which may already be significant, although only a small proportion of the state transitions has been searched. Similarly, here a histogram of stable sets could be made. The sampling version of the search may still have the advantage over simulations that it is independent of the initial conditions.

## Conclusion

The artificial chemistries with boolean networks allow the study of self-replicating sets. Since no qualitative differences between Dittrich and Banzhaf's (1998) automaton reaction and the boolean networks have been found, the use of boolean networks has allowed to use shorter strings. Simulations have shown a high sensitivity to the filter rate, and that different stable sets can result as a consequence of such little changes. The search finds the different stable sets more efficiently, at least for short strings, because it is independent of the initial population and the filter rate.

In addition, the search method may be extendable to a reverse characterisation of initial conditions when the stable set is given. Initial conditions may be characterised by containing sufficient support for a set, or the decision boundary of two competing sets may be expressed by the sizes of certain subpopulations.

The current results suggest the following answers to the questions raised in the introduction:

1. (Number and size of autocatalytic sets?) The search gives the number of potential sets and their reaction subgraphs, which can be classified into types. All potential sets including the all-string set require testing for their dynamical stability. Finding the number of autocatalytic sets was motivated by the possibility that artificial chemistries could be considered as adaptive systems. However, a small number of autocatalytic sets indicates too little freedom for adaptation.
2. (Autocatalytic sets related to subgraphs?) The closed subgraphs are a good indicator, but not necessary for the existence of autocatalytic sets (see above discussion of parasites). However, according to the two criteria in the section "A search method", the number of sets that are found is only limited by the number of combinations in the starting sets. It appears that only very peculiar types of autocatalytic reaction patterns would require starting sets much greater than three strings.
3. (Boolean nets compared to Dittrich and Banzhaf's automaton?) As in Dittrich and Banzhaf's (1998) simulation, there is first an exploration phase before any autocatalytic set is found. The sets that are found can have more than one member, so the boolean network does not give rise to different characteristics. For some boolean networks very many potential sets are found, which seems to be an interesting property regarding the short string length of five bits. However, the very many types of reaction patterns resulting from some searches still require more investigation. It may be possible that trends are found that hold for longer string lengths as well.

Since the behaviour has been found rather fixed, it seems not to be universal enough for attempting adaptation to external constraints. Applying some external fitness can only select from the rather small number of possible stable sets.

Rather than attempting universal adaptation, this type of system may serve for studying evolution. Boolean network may be selected that exhibit bigger sets where the dynamics is not stationary (e.g. like the waves observed by Dittrich, Ziegler & Banzhaf (1998)), or other influences like mutations or interactions with an environment may be added. The description of the system resulting from the search can facilitate a quick comparison of different systems (as characterised by RAM connections, memory assignment *and* initial populations according to the possible sets found).

A subject of future research may be to inject new species into the population, to consider several interacting spatial compartments, or to study if predictions can be made about consequences of any changes of the boolean network parameters (RAM connection, memory assignment). In the case of injecting species into the population, the current results can help to characterise the *reachability* of stationary states from given initial conditions.

### Acknowledgements

Many thanks to W. Banzhaf, P. Dittrich, and J. Ziegler for helpful discussions.

This work was supported by the European Commission, in the programme Training and Mobility of Researchers.

### References

- Aleksander, I. & Morton, H. *An Introduction to Neural Computing*, Chapman & Hall 1990.
- Bagley, R. J., Farmer, J. D. & Fontana, W. (1992). Evolution of a Metabolism. In C. G. Langton, C. Taylor, J. D. Farmer & S. Rasmussen (Eds.) *Artificial Life II*. Santa Fe Institute Proceedings Vol. X, (pp. 141-158). Redwood City, CA: Addison-Wesley.
- Dittrich, P. & Banzhaf, W. (1998). Self-Evolution in a Constructive Binary String System. *Artificial Life* 4(2), 203-220.
- Dittrich, P., Ziegler, J. & Banzhaf, W. (1998). Mesoscopic Analysis of Self-Evolution in an Artificial Chemistry. Proc. of *Artificial Life VI*, Los Angeles, June 26-29, 1998; C. Adami, R. Belew, H. Kitano, and C. Taylor (Eds.), MIT Press, pages 95-103.
- Eigen, M. & Schuster, P. (1978). The Hypercycle. A Principle of Natural Self-Organization. Part B: The Abstract Hypercycle. *Die Naturwissenschaften* 65, 7-41.
- Farmer, J. D., Kauffman, S. A. & Packard, N. H. (1986). Autocatalytic replication of polymers. *Physica D* 22, 50-67.
- Hünig, H. (2000). Convergence Analysis of a Segmentation Algorithm for the Evolutionary Training of Neural Networks. To appear at: *The First IEEE Symposium on Combinations of Evolutionary Computation and Neural Networks*. May 11-12, 2000, San Antonio, TX, USA.
- Ikegami, T. & Hashimoto, T. (1995). Coevolution of Machines and Tapes. In F. Moran et al. (Eds.) *Advances in Artificial Life*. LNAI 929, pp. 234-254.
- Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theoret. Biol.* 22, 437-467.
- Stadler, P. F., Fontana W. & Miller, J. H. (1993). Random catalytic reaction networks. *Physica D* 63, 378-392.
- Wuensche, A. (1997). Attractor Basins of Discrete Networks. PhD Thesis, The University of Sussex, Brighton, UK.