in: Proceedings des Workshops "Artificial Life"
Kerstin Dautenhagn et al., Editors,
pp. 29-38, GMD-Studien Nr. 271, Sankt Augustin, 1995

Selbstevolution in einem Binärstringsystem

PETER DITTRICH[1] und WOLFGANG BANZHAF[2]

Fachbereich Informatik, Universität Dortmund [3]
D-44221 Dortmund

Zusammenfassung

Dieser Beitrag stellt ein algorithmisches Reaktionssystem von interagierenden 32-Bit-Binärstrings vor. Es basiert auf der chemischen Metapher der Informationsverarbeitung. Als besonderes beobachtetes Phänomen soll hier das der Evolution herausgegriffen werden. Sein Auftreten ist insofern bemerkenswert, als keine expliziten Mutations- und Rekombinationsoperatoren oder künstliche Selektionsoperatoren eingesetzt werden. Jegliche Variation wird durch die Spezies selbst bzw. durch ihre Sekundärstruktur in Form endlicher Automaten vorgenommen.

Stichworte: Selbstorganisation, präbiotische Evolution, molekulare Computer
Selbstprogrammierung

Einleitung

Die Frage nach der Entstehung des Lebens und der Entwicklung der beeindruckenden Vielfalt und Sinnhaftigkeit seiner Natur ist weiterhin unbeantwortet. Die abermillionenfache Dimensionalität des Variantenraums offenbart, daß eine blinde Suche, getrieben vom Darwinschen Prinzip der Variation und Selektion, als Erklärung unzureichend ist. Aktuelle Erklärungsversuche inkorporieren Konzepte der Selbstorganisation und der spontanen Strukturbildung (z.B. [KAUFFMAN 1993]). Die Validierung solcher Theorien ist problematisch, da keine brauchbaren Fragmente der präbiotischen Evolution die Jahrmillionen überdauert haben. Auch Rekonstruktionsversuche im Labor haben bisher keine befriedigenden Ergebnisse geliefert.

Abhilfe können Computerexperimente bieten. Allerdings wird nicht eine detaillierte Simulation der physikalischen Vorgänge angestrebt, sondern die Instanziierung einer künstlichen Welt in einem Computersystem, wobei die dort vorhandenen "natürlichen" Fähigkeiten respektiert und effizient genutzt werden [RAY 1991]. Dieses Vorgehen macht nur Sinn, falls die grundlegende Arbeitshypothese stimmt, daß die für lebende Systeme typischen Phänomene unabhängig von ihrer materiellen Basis sind [LANGTON 1989].

Die "materielle" Basis in dieser Arbeit sind endliche Sequenzen von Nullen und Einsen – sogenannte **Binärstrings** – von konstanter Länge 32. Die Welt, in der sie agieren besteht aus einer Population von M Binärstrings. Zwei Strings können miteinander kollidieren und dabei einen Neuen bilden. Dazu wird ein String in seine **Sekundärstruktur** überführt, die einer Maschine entspricht, welche den anderen String als Eingabe verarbeitet und das Reaktionsprodukt als Ausgabe liefert. Zur Beschreibung der Sekundärstruktur wird ein endlicher formaler Automat verwendet, dessen Funktionalität eindeutig durch die Primärstruktur des Strings determiniert ist.

[1] e-mail: dittrich@LS11.informatik.uni-dortmund.de
[2] e-mail: banzhaf@LS11.informatik.uni-dortmund.de
[3] URL: http:\\ls11-www.informatik.uni-dortmund.de

Aufbau

Das verwendete Experimentier-System, das man als einen **algorithmischen Reaktor** bezeichnen könnte, besteht im Wesentlichen aus drei Komponenten: 1.) Einer **Population von Objekten**. 2.) Einen **Reaktionsmechanismus**, der die Wechselwirkung zwischen den Objekten beschreibt. 3.) Einen **Algorithmus**, der für den Betrieb des Systems verantwortlich ist. Änliche Systeme werden unter anderem von FARMER, KAUFFMAN, UND PACKARD (1986); FONTANA (1991, 1994); THÜRK (1993); BANZHAF (1993, 1994, 1995) untersucht.

Die Population von M Binärstrings $s \in S = \{0,1\}^{32}$ wird in dieser Arbeit als räumlich homogen betrachtet. Das heißt, daß kein String eine ausgezeichnete Position hat, und der Zustand der Population als Konzentrationsvektor $x(s) = n(s)/M$ dargestellt werden kann mit $n(s) =$ "Anzahl des Strings s in der Population".

Der eingesetzte Reaktionsmechnismus – die **Automaten-Reaktion** – läßt sich als chemische Reaktionsgleichung notieren:

$$s + s' + X \longrightarrow s + s' + s''$$

oder kürzer: $$s + s' \Longrightarrow s''$$

Die beiden Reaktanden s, s' bleiben bei der Reaktion unverändert und wirken so als Katalysatoren bei der Erzeugung von s''. Das "Substrat" X dient nur zur Erhaltung der Stoffbilanz und taucht in dem System nicht explizit auf. Es könnte als eine Rechnerresource (z.B. Speicherplatz, Rechenzeit) interpretiert werden.

Der "Betrieb" erfolgt durch den Reaktor-Algorithmus. Er simuliert einen gut durchmischten Reaktionsbehälter konstanter Größe.

Reaktor-Algorithmus

- *Initialisierung (Befüllung des Reaktors):*
  Erzeuge M zufällige Objekte aus S. Diese bilden die Population.
- *Iteration (Betrieb des Reaktors):*
  1. *Wähle zufällig zwei Objekte s, s' aus der Population aus, ohne diese zu entfernen.*
  2. *Bilde ihr Reaktionsprodukt $s + s' \Longrightarrow s''$.*
  3. *Falls die Filterbedingung $f(s, s', s'')$ nicht erfüllt ist, ersetze zufällig ein Objekt der Population durch s''.*

Ein Filter wird hier zur Einführung eines **Replikationsverbots** verwendet. Die Filterbedingung $f(s, s', s'')$ ist dabei erfüllt, falls $s = s''$ oder $s' = s''$ gilt. M Iterationen heißen eine **Generation**. Dabei handelt es sich um eine Laufzeitgröße, die unabhägig von der Populationsgröße M ist. Für spezielle Fälle kann das Verhalten des Systems mit gewöhnlichen nicht-linearen Differntialgleichungen modelliert werden [EIGEN UND SCHUSTER 1979; HOFBAUER UND SIGMUND 1984; STADLER, FONTANA, UND MILLER 1993].

Unter der (momentanen) **Produktivität** versteht man die Wahrscheinlichkeit, daß bei einer Iteration ein Objekt in die Population eingefügt wird. Wird keine Filterbedingung verwendet, so ist die Produktivität konstant gleich Eins. Zur Messung der Produktivität werden die Anzahl der eingefügten Objekte pro Generation gezählt. Als ein Maß für die **Diversität** einer Population wird im folgenden die absolute Anzahl unterschiedlicher Objekte verwendet.

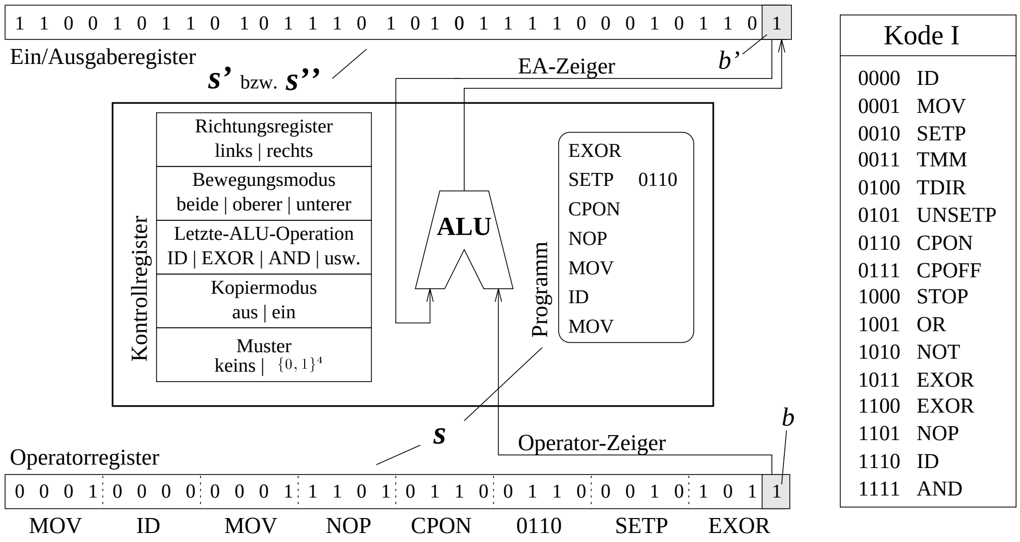


Abb. 1: *Schema des Automaten, der aus der Faltung von s entsteht. Er realisiert die Reaktion $s + s' \Longrightarrow s''$. s wird zu Beginn in das Operatorregister geschrieben und legt das Programm fest. Das Ein/Ausgaberegister wird mit s' initialisiert und enthält nach Abarbeitung des Programms das Resultat s''.*

Die Automaten-Reaktion

Die **Automaten-Reaktion** realisiert eine eindeutige Reaktion $s + s' \Longrightarrow s''$ zwischen zwei Binärstrings der Länge 32 bit. Dazu wird zunächst s zu einem Automaten A_s gefaltet, der dann mit s' als Eingabe gestartet wird. Er ist so konstruiert, daß er nach endlicher Zeit hält. Seine Ausgabe $A_s(s')$ entspricht s''. Da es sich bei A_s um einen *deterministischen* Automaten handelt, definiert die Automaten-Reaktion eine Funktion $\{0,1\}^{32} \times \{0,1\}^{32} \to \{0,1\}^{32}$.

Der Automat verfügt über zwei 32-Bit-Register (Abb. 1). Das **Operatorregister** wird mit s geladen und bestimmt das **Steuerprogramm** des Automaten. Es kann nicht verändert werden. Das **Ein/Ausgaberegister** wird mit s' initialisiert. Wird der Automat gestartet, so wird das Steuerprogramm Schritt für Schritt abgearbeitet. Nach dem letzten Befehl terminiert der Automat. Es gibt keine Kontrollkonstrukte, wie zum Beispiel Schleifen oder Sprünge.

Für jedes Register markiert ein Zeiger eine Bit-Position. Die **ALU** kann diese beiden Bits miteinander verknüpfen (z.B. durch ein logisches UND). Das Resultat – ebenfalls 1 Bit – wird in das Ein/Ausgaberegister geschrieben. Die Zeiger können auf vielfältige Art verschoben werden.

Die Tabelle 1 veranschaulicht die Funktionsweise des Automaten anhand des Beispiels

$$101d662b + cb5d5c2d \Longrightarrow cb5d5606 \qquad \text{(hexadezimale Darstellung)}$$

(vgl. Abb. 1). Für eine ausführliche Darstellung sei auf [Dittrich 1995] verwiesen.

Versuchsdurchführung und Ergebnisse

Im folgenden werden drei Versuchsreihen beschrieben. Von Reihe zu Reihe wird die Schwierigkeit, daß sich eine stabile, selbsterhaltende Organisation ausbildet, erhöht, indem zunächst die Replikation verboten und schließlich die NOT-Operation entfernt wird. In dieser dritten

s	Befehl	Kommentar
1011	EXOR	Das Bit, auf das der Operator-Zeiger deutet, und das Bit auf das der EA-Zeiger weist werden Exklusiv-Oder verknüpft und das Resultat (0) an die Stelle des EA-Zeigers geschrieben, so daß das 0-te Bit des Ein/Ausgaberegisters den Wert 0 hat. Danach werden beide Zeiger um eine Position nach links verschoben.
0010 0110	SETP 0110	Setzt das Musterregister des Automaten auf den Wert 0110. Bei einer folgenden MOV-Operation werden die Zeiger solange bewegt, bis das Muster im Operatorregister gefunden wird (max. 32 Schritte).
0110	CPON	Schaltet den Kopiermodus ein. Bei jeder nun folgenden MOV-Operation wird vor jedem Bewegungsschritt der Zeiger die letzte ALU Operation (hier: EXOR) durchgeführt.
1101	NOP	Keine Operation. Die Register des Automaten bleiben unverändert.
0001	MOV	Bewegung der Zeiger. Diese hängt von Zustandsregistern, wie zum Beispiel dem Musterregister oder dem Bewegungsregister ab. Hier werden beide Zeiger solange um eine Position verschoben, bis das Muster 0110 im Operatorregister gefunden ist; also bis zum 8. Bit. Da der Kopiermodus aktiviert worden ist (mit CPON), wird während der Bewegung die letzte ALU Operation (hier: EXOR) für jedes überschrittene Bit ausgeführt.
0000	ID	Das 8. Bit des Operatorregisters wird in das Ein/Ausgaberegister kopiert, und beide Zeiger werden einen Schritt nach links bewegt (ID = Identität).
0001	MOV	Bewegung beider Zeiger zum nächsten Muster 0110, auf die 12. Position. Diesmal wird in jedem Schritt ein Bit vom Operatorregister in das Ein/Ausgaberegister kopiert.

Tabelle 1: *Beispiel für die Arbeitsweise der Automaten-Reaktion (vgl. Abb. 1).*

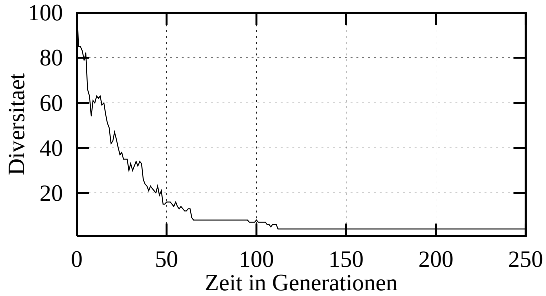
Versuchsreihe, C, zeigt das System evolutives Verhalten. Bei allen drei Versuchsreihen wird die Anfangspopulation mit M zufälligen Binärstrings aus S initialisiert.

Die sich ausbildenden Organisationsstrukturen sind den von FONTANA beobachteten sehr ähnlich. Im Gegensatz zu den Untersuchungen in [FONTANA 1991; FONTANA UND BUSS 1994] steht hier jedoch nicht die statische Struktur einer Organisation im Vordergrund, sondern der qualitative Verlauf ihres Entstehungsprozesses, also die Dynamik.

Reihe A: Experimente mit Replikation

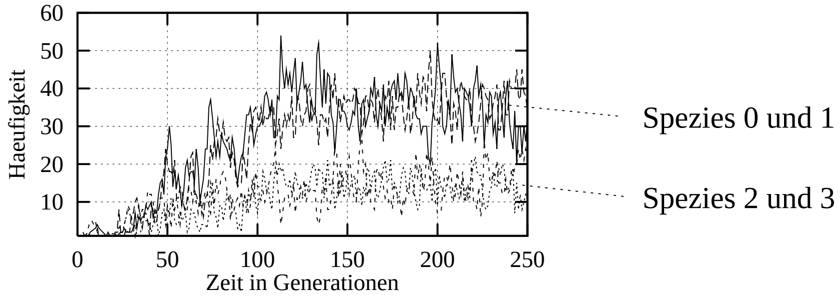
Die Abbildung 2 zeigt einen typischen Lauf des Reaktor-Algorithmus ohne Filterbedingung mit einer Populationsgröße von $M = 100$. Am Ende hat sich die Diversität auf 4 verschiedene Stringsorten reduziert, die eine stabile selbsterhaltende Organisationsstruktur bilden. Typisch für derartig kleine Populationen ist, daß die am Ende dominierenden Spezies praktisch schon von Anfang an auftreten. Wenn sie nicht in der Anfangspopulation vorhanden sind, so entstehen sie in den ersten Generationen. Der Reaktor ist nicht sonderlich innovativ.

Bei großen Populationen ($M > 5000$) entsteht mit hoher Wahrscheinlichkeit ein **fleißiger Replikator** s, für den gilt: $\forall s' \in S : s + s' \Longrightarrow s$. Dieser ist also in der Lage, sich unabhängig vom gewählten Operanden s' zu replizieren. Ein solcher String wirkt als **Destruktor** und verdrängt alle anderen Sorten, so daß die Diversität gegen Eins läuft.



Reaktionsmatrix der letzten vier Spezies (ab Gen. 122)

Nr.	Spezies	0	1	2	3
0	6266edbb	2	3	0	1
1	6266edb9	0	1	1	0
2	6266edb8	0	1	2	3
3	6266edba	1	0	0	1



Spezies 0: 6266edbb

1011	EXOR
1011	EXOR
1101	NOP
1110	ID
0110	CPON
0110	CPON
0010 0110	SETP 0110

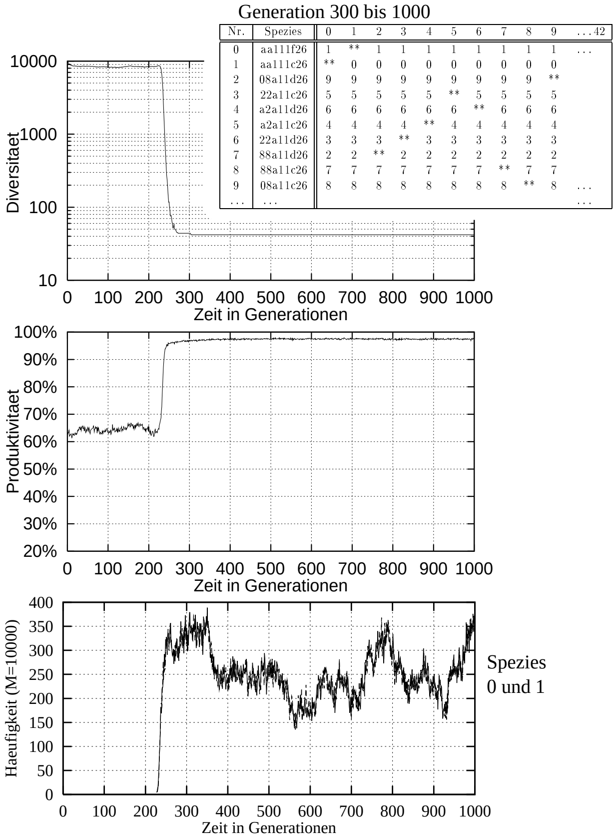
Spezies	1. Halbbyte	1. Befehl
0	b = 1011	EXOR
1	9 = 1001	OR
2	8 = 1000	NOP
3	a = 1010	NOT

Abb. 2: *Typischer Lauf des Reaktor-Algorithmus ohne Replikationsverbot und einer Populationsgröße von $M = 100$. Die Diversität gibt die absolute Anzahl unterschiedlicher Spezies in der Population an. Die starken rauschartigen Schwankungen der Graphen sind auf die nichtdeterministischen Elemente des Reaktor-Algorithmus zurückzuführen. Sie machen sich besonders bei kleinen Populationen bemerkbar. Die Reaktionsmatrix ist eine kompakte Schreibweise der Rekationsgleichungen $s + s' \Longrightarrow s''$ für die angegebenen Spezies. Sie sind nach Häufigkeit geordnet. Nr. 0 ist der häufigste String der Population.*

Reihe B: Experimente ohne Replikation

Verbietet man die Replikation, so wird sowohl das Verhalten, als auch die sich ausbildende Organisationsstrukturen komplexer. Eine stabile Organisation muß aus einem Netzwerk kooperierender Spezies bestehen. Die Abbildung 3 zeigt einen typischen Lauf des Reaktor-Algorithmus mit Replikationsverbot und $M = 10^4$. Der Entstehungsprozeß unterscheidet sich qualitativ deutlich von dem der Abbildung 2. Betrachtet man den Verlauf der Diversität, so erkennt man drei Phasen: In der **innovativen Phase** (Generation 0-225) herrscht eine hohe Diversität. Laufend werden neue Spezies gebildet, die nicht in der Population vertreten sind. Keine Spezies ist außergewöhnlich häufig. Dabei führt die intensive Exploration des Variantenraums zur Entstehung von Organisationsstrukturen, die sich als Ganzes wie fleißiege Replikatoren verhalten, wie zum Beispiel:

$$\begin{array}{lll} s_0 + s' \Longrightarrow s_1 & s_1 + s' \Longrightarrow s_0 & \text{oder} \\ s_2 + s' \Longrightarrow s_9 & s_9 + s' \Longrightarrow s_8 & \\ s_8 + s' \Longrightarrow s_7 & s_7 + s' \Longrightarrow s_2 & \end{array}$$



Nr.	Spezies	0	1	2	3	4	5	6	7	8	9	...42
0	aa111f26	1	**	1	1	1	1	1	1	1	1	...
1	aa111c26	**	0	0	0	0	0	0	0	0	0	
2	08a11d26	9	9	9	9	9	9	9	9	9	**	
3	22a11c26	5	5	5	5	5	**	5	5	5	5	
4	a2a11d26	6	6	6	6	6	6	**	6	6	6	
5	a2a11c26	4	4	4	4	**	4	4	4	4	4	
6	22a11d26	3	3	3	**	3	3	3	3	3	3	
7	88a11d26	2	2	**	2	2	2	2	2	2	2	
8	88a11c26	7	7	7	7	7	7	7	**	7	7	
9	08a11c26	8	8	8	8	8	8	8	8	**	8	...
...	...											...

Abb. 3: *Typischer Lauf des Reaktor-Algorithmus mit Replikationsverbot ($M = 10^4$). Die am Ende dominierenden Spezies sind nicht von Anfang an vorhanden, sondern sind – im Gegensatz zur Abbildung 2 – das Ergebnis einer langen innovativen Phase hoher Diversität. Die Produktivität ist der Anteil der reaktiven Iterationen einer Generation. Das Symbol "**" in der Reaktionsmatrix bezeichnet eine elastische Wechselwirkung aufgrund des Replikationsverbots.*

wobei s' eine beliebige Spezies aus der Population ist. Mit dem Auftauchen derartiger Binärstrings setzt eine **selektive Phase** ein, in der eine Verdrängung anderer Stringsorten stattfindet, die sich nicht wie zuvor kooperativ selbsterhalten können. Die Diversität fällt rapide. Am Ende der selektiven Phase ist die Organisation **abgeschlossen**. Jede Kollision zweier Strings führt zu einem String, der schon vorhanden ist. Es folgt eine **stagnierende Phase** geringer Diversität, in der nichts Neues entstehen kann.

Die "Standardmethode" zur Umgehung des Replikationsverbots, die sich in den Simulationen mit $M >= 10^4$ entwickelt hat und die auch die Spezies der Abbildung 3 anwenden, enthält zwei Schritte: 1.) Das Kopieren des Operatorregisters in das Operandenregister. 2.) Das Einbringen einer geringen Variation durch die Invertierung von wenigen Bits im Operandenregister mittels der NOT-Operation. Die resultierende Spezies bewirkt dasselbe und macht diese Variation wieder rückgängig, falls sie als Operator bei einer Reaktion fungiert.

Reihe C: Experimente ohne Replikation und ohne NOT

Enfernt man die NOT-Operation, die in der Versuchreihe B von den Spezies zur Umgehung des Replikationsverbots verwendet wurde, so erhöht sich nochmals die Schwierigkeit eine zusammenhängende, selbsterhaltende Organsiationsstruktur zu bilden. Abbildung 4 zeigt die

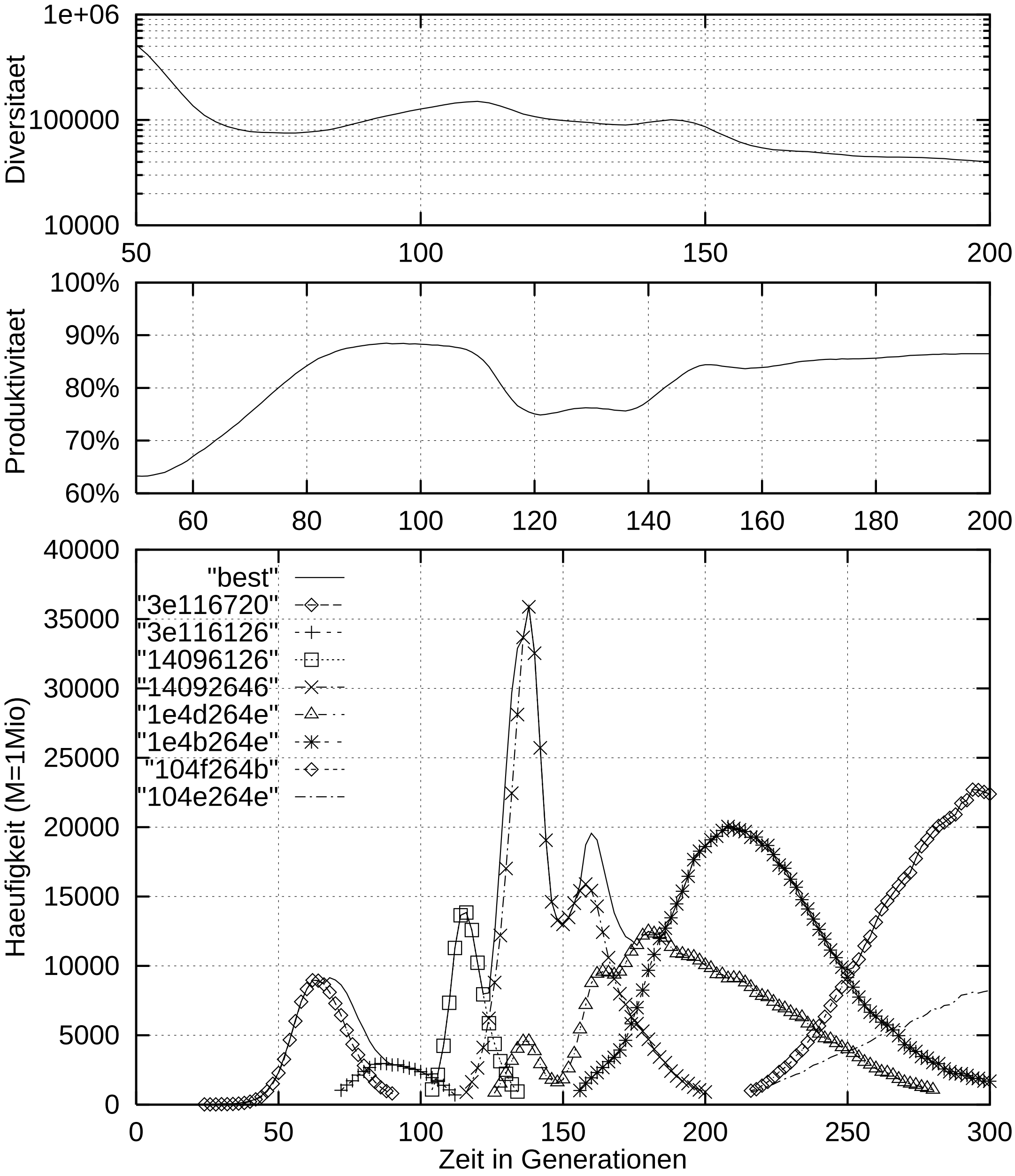


Abb. 4: *Evolutive Phase eines Laufs mit einer Populationsgröße von $M = 10^6$. Die dargestellten Häufigkeitsverläufe einzelner Spezies stehen stellvertretend für ganze Gruppen von Binärstrings, die auftauchen und wieder verschwinden. Der Entwicklungsprozess ist auch in der Struktur der Strings zu erkennen. Jede Sequenz hat dabei einen Teil ihres "Vorgängers" geerbt. Der Graph "best" gibt die größte Häufigkeit einer Spezies an. (Reihe C, ohne Replikation und ohne NOT-Operation)*

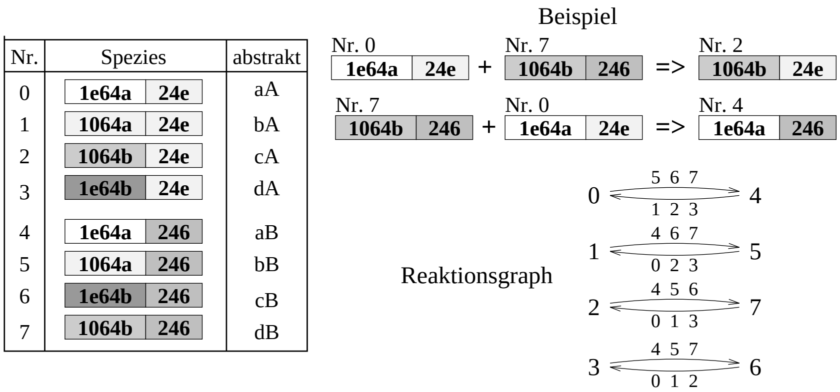


Abb. 5: *Crossover-Methode zur Umgehung des Replikationsverbots. Der Operator kopiert nur einen Teil seiner eigenen Binärsequenz in das Operandenregister, so daß das Produkt aus Teilsequenzen der Reaktanden zusammengesetzt ist. Im Reaktionsgraph rechts unten sind zur Verbesserung der Übersicht die Operatoren über den jeweiligen Reaktionspfeil geschrieben. Die Spezies sind eng gekoppelt, obwohl die Darstellung etwas anderes suggeriert.*

ersten 300 Generationen eines Laufs mit einer sehr großen Population von $M = 10^6$. Dabei ist die NOT-Operation durch die EQ-Operation (Gleichheit: $b' := (b = b')$) ersetzt worden. Typisch für die Reihe C ist, daß sich die am Ende dominierende Organisation nicht plötzlich bildet, sondern das Ergebnis eines Entwicklungsprozesses ist. Die Entwicklungsschritte sind durch das Auftauchen einer Gruppe von Spezies in hoher Konzentration und das baldige – oft vollständige – verschwinden derselben gekennzeichnet. Die Abbildung zeigt für einige dieser Gruppen exemplarisch ihre jeweils häufigsten Stellvertreter. Betrachtet man die Folge dieser Spezies, so ist zu erkennen, daß jede Spezies Teilsequenzen ihres Vorgängers "geerbt" hat.

Unter der Abwesenheit der NOT-Operation hat sich eine andere Methode zur Umgehung des Replikationverbots entwickelt als in der Reihe B. Sie ist dadurch gekennzeichnet, daß Reaktionsprodukte aus Teilsequenzen der Reaktionspartner zusammengesetzt werden (Abb. 5). Die entsprechenden Spezies sind in der Lage, ein Stück ihrer eigenen Binärsequenz in den Kode einer anderen einzupflanzen. Dies wird deutlich, wenn man sie mit den Spezies `00000000` und `ffffffff` reagieren läßt:

Dominante Spezies eines Laufs der Reihe C	Spezies s	$s +$ `00000000`	$s +$ `ffffffff`
häufigste Spezies der Gen. 998	`1e64a24e`	`0000024e`	`fffff24e`
5. häufigste Spezies der Gen. 998	`1e64a246`	`0000024e`	`fffff24e`
176. häufigste Spezies der Gen 998	`1064f204`	`00000204`	`7ffff204`
200. häufigste Spezies der Gen. 998	`1e6c524c`	`1e6c4000`	`1e6c5ffe`

Diskussion

Die Experimente haben drei qualitativ unterschiedliche Verläufe von Selbstorganisationsvorgängen gezeigt. In der Versuchsreihe C konnte die Verhaltensvielfalt durch die Entfernung einer "wichtigen" Operation (NOT) gesteigert werden. Die Spezies werden "herausgefordert", eine neue Methode selbständig zu entwickeln, um das Replikationsverbot ohne den Einsatz der NOT-Operation zu umgehen.

Der Entstehungsprozess der am Ende dominierenden Organisation hat den Charakter einer präbiotischen Evolution (vgl. [HOFBAUER UND SIGMUND 1984] Abb. 15.3). Im Gegensatz zur Darwinschen Evolution, bei der sich unterschiedliche Arten von Spezies mit einem weit verzweigten Stammbaum entwickeln, hat sich hier bei keiner Simulation eine Organisation gebildet, in der Spezies mit grundsätzlich unterschiedlichem Aufbau stabil koexistieren konnten. Dieses Phänomen ist vermutlich auf die räumlich homogene Struktur der Population zurückzuführen. Es gibt keine Nischen, in denen sich verschiedene Speziesarten entwickeln können.

Das System ist zu einer "natürlichen" Evolution fähig, ohne daß externe, künstliche Operatoren für Replikation, Reproduktion, Mutation, Kreuzung oder Selektion, die bezüglich einer vorgegebenen Fitness erfolgt, eingesetzt werden. Allerdings findet eine implizite Selektion statt, die aufgrund der Kompetition um den beschränkten Reaktorraum entsteht.

Dies steht im Gegensatz zu einem Evolutionärem Algorithmus (EA), bei dem die Individuen als passive Träger der genotypischen Information behandelt werden und die Variationen durch explizite Mutations- und Rekombinationsoperatoren erfolgen. Aus der Sicht der Population erfolgt bei einem EA eine Fremdorganisation durch diese Operatoren. Da hier der Algorithmus zu Veränderung der Spezies durch sie selbst repräsentiert wird und keine Variation von außen eingebracht wird, soll das Phänomen als **Selbstevolution** bezeichnet werden.

Ein interessanter Aspekt eines selbstevolutiven Systems ist der folgende Punkt:

Dadurch, daß die Spezies sich selbst verändern, verändern sie auch die Art und Weise, *wie* diese Modifikation erfolgt. Besonders deutlich ist dies bei der Entstehung des Crossover-Mechanismus in der Versuchsreihe C geworden, der eine Veränderung in der Evolutionsfähigkeit der Spezies bewirkt. Ein solches Verhalten kann man als Meta-Evolution bezeichnen. Da die Art der Meta-Evolution ebenfalls durch die Spezies bestimmt wird, ist auch eine Veränderung der Meta-Evolutionsfähigkeit denkbar und damit auch eine Meta-Meta-Evolution usw. Eine anscheinend endlose "seltsame" Schleife. Die Schleife entsteht hier durch die Operator-Operanden-Dualität der Spezies und den dadurch ermöglichten Selbstbezug.

Hat hier eine Meta-Evolution oder gar eine Meta-Evolution noch höherer Ordnung stattgefunden? Sicher ist, daß eine *Veränderung* der Evolutionsfähigkeit erfolgt ist. Für den Nachweis einer *Evolution* der Evolutionsfähigkeit müßten mehrere gerichtete Schritte ihrer Entwicklung aufgezeigt werden. Aufgrund der beschränkten Größe der Binärstrings und der beobachteten relativ raschen Reduzierung der Vielfalt ist eine Meta-Evolution noch höherer Ordnung äußerst unwahrscheinlich.

Der für die Wechselwirkung definierte Automat besteht aus Elementen gängiger Computer. Er läßt sich leicht als Gatterschaltung realisieren. Es liegt deshalb die Vermutung nahe, daß nicht nur der hier verwendete Automat, sondern auch reale Rechnersysteme eine "natürliche" intrinsische Fähigkeit zur Selbstorganisation bzw. "Selbstprogrammierung" besitzen.

Der Reaktor-Algorithmus realisiert ein Modell eines chemischen Reaktors. Kürzlich erschienene Arbeiten haben gezeigt, wie Reaktionssysteme zur Informationsverarbeitung und -speicherung eingesetzt werden können [HJELMFELT, WEINBERGER, UND ROSS 1991; ADLEMAN 1994; DITTRICH 1995; DITTRICH UND BANZHAF 1995]. Eine Integration der hier gezeigten Entwicklungsfähigkeit in derartige Ansätze erscheint vielversprechend.

Als zukünftige Arbeiten ist die Entwicklung eines geeigneten Instrumentariums zur Beobachtung der Population und die Einführung einer räumlichen Struktur geplant.

Literatur

Adleman, L. M. (1994). Molecular Computation of Solutions to Combinatorial Problems. *Science 266*, 1021–1024.

Banzhaf, W. (1993). Self-replicating sequences of binary numbers – Foundations I and II: General and Strings of Length N = 4. *Biological Cybernetics 69*, 269–281.

Banzhaf, W. (1994). Self-organisation in a system of binary strings. In R. Brooks und P. Maes (Eds.), *Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems.* MIT Press, Cambridge MA.

Banzhaf, W. (1995). Self-Organizing Algorithms Derived from RNA Interaction. In W. Banzhaf und F. H. Eeckman (Eds.), *Evolution and Biocomputation: Computational Models of Evolution*, S. 69–102. Springer, Berlin-Heidelberg-New York.

Dittrich, P. (1995). Selbstorganisation in einem Binärstringsystem mit algorithmischen Sekundärstrukturen. Diplomarbeit, Universität Dortmund, Fachbereich Informatik, Lehrstuhl für Systemanalyse.

Dittrich, P. und W. Banzhaf (1995). The Chemical Computation Metaphor and Robot Control. ECAL 95, Posterbeitrag.

Eigen, M. und P. Schuster (1979). *The Hypercycle.* Springer, Berlin.

Farmer, J. D., S. A. Kauffman, und N. H. Packard (1986). Autocatalytic replication of polymers. *Physica D 22*, 50–67.

Fontana, W. (1991). Algorithmic Chemistry. In C. G. Langton et al. (Eds.), *Artificial Life II: Proceedings of the Second Artificial Life Workshop*, S. 159–209. Addison-Wesley, Reading MA.

Fontana, W. (1994). Molekulare Semantik - Evolution zwischen Variation und Konstruktion. In V. Braitenberg und I. Hosp (Eds.), *Entwicklung und Organisation in der Natur*, S. 69–106. Rowohlt, Hamburg.

Fontana, W. und L. W. Buss (1994). "The Arrival of the Fittest": Toward A Theory of Biological Organisation. *Bull. Math. Biol. 56*, 1–64.

Hjelmfelt, A., E. D. Weinberger, und J. Ross (1991). Chemical implementation of neural networks and Turing machines. *Proc. Natl. Acad. Sci. USA 88*, 10983–10987.

Hofbauer, J. und K. Sigmund (1984). *Evolutionstheorie und dynamische Systeme.* Paul Parey, Berlin.

Kauffman, S. A. (1993). *The Origins of Order.* Oxford University Press, New York.

Langton, C. G. (1989). Artificial Life. In C. G. Langton (Ed.), *Artificial Life*, S. 1–45. Addison-Wesley, Redwood City.

Ray, T. S. (1991). An Approach to the Synthesis of Life. In C. G. Langton, C. Taylor, J. D. Farmer, und S. Rasmussen (Eds.), *Artificial Life II: Proceedings of the Second Artificial Life Workshop*, S. 371–408, Addison-Wesley, Reading MA.

Stadler, P. F., W. Fontana, und J. H. Miller (1993). Random catalytic reaction networks. *Physica D 63*, 378–392.

Thürk, M. (1993). *Ein Modell zur Selbstorganisation von Automatenalgorithmen zum Studium molekularer Evolution.* Doktorarbeit, Universität Jena, Naturwissenschaftliche Fakultät.