

**Eine Untersuchung komplexer Anwendungen
zellulärer Automaten in Hinblick auf Möglichkeiten
innovativer musikalischer Strukturgenese**

Toningenieur-Projekt
durchgeführt von

Jamilla Balint

Institut für elektronische Musik und Akustik
der Universität für Musik und darstellende Kunst Graz

Betreuer: Mag. art. Dr. phil. Gerhard Nierhaus

Graz, im Jänner 2014

Zusammenfassung

Zellulare Automaten sind dynamische Systeme im diskreten Raum. Dieser Raum wird in einzelne Zellen unterteilt, welche in Abhängigkeit benachbarter Zellen verschiedene Zustände annehmen können. Diese Zustandsänderungen erfolgen für alle Zellen des Automaten gemeinsam in zeitdiskreten Schritten. Einfache Ausgangsbedingungen führen zu einem komplexen Systemverhalten, das auch zur Untersuchung und Simulation von Phänomene wie der Ausbreitung von Gasen und Flüssigkeiten, Reibungseffekte an Oberflächen oder das Schwarmverhalten von Tieren verwendet werden kann. Die gegenständliche Arbeit abstrahiert von den gängigen einfachen Mappingstrategien in der musikalischen Strukturgenese und konzentriert sich auf komplexere Anwendungen zellulärer Automaten, wie sie bspw. in der Medizin, Physik und Biologie Verwendung finden. Daraus resultierend sollen komplexe Mappingstrategien aus der Wissenschaft in Hinblick auf ihre Anwendbarkeit in der musikalischen Strukturgenese untersucht werden.

Inhaltsverzeichnis

1	Einführung	4
1.1	Die Arbeit im Überblick	6
1.2	Definitionen	6
1.2.1	Zellraum	6
1.2.2	Nachbarschaft	7
1.2.3	Randbedingungen	8
1.2.4	Zustandsmenge	9
1.3	Typen von zellularen Automaten	9
1.3.1	1-dimensionaler Automat	9
1.3.2	2-dimensionaler Automat	11
1.3.3	3-dimensionaler Automat	11
2	Historischer Überblick	14
2.1	Das Ising-Modell	15
2.1.1	Das Ising-Modell als zellulärer Automat	16
2.2	John von Neumann	
	The Theory of Self-Reproducing Automata	16
2.3	Konrad Zuse - Rechnender Raum	18
2.3.1	Der zellulare Automat im Rechnenden Raum	18
2.4	John Conway - Game of Life	19
2.4.1	Organismen in GoL	21
2.5	Stephen Wolfram - A New Kind of Science	25
2.5.1	Klasse 1	27
2.5.2	Klasse 2	27
2.5.3	Klasse 3	27
2.5.4	Klasse 4	28

2.5.5	Häufigkeit der Klassen im 1-dimensionalen Raum	28
2.5.6	Der Langton-Parameter	28
3	Zelluläre Automaten in der Wissenschaft	30
3.1	ZA in der Verkehrsmodellierung	30
3.1.1	Das Nagel-Schreckenberg-Modell	30
3.1.2	Das Behine-Middleton-Levine-Modell	31
3.1.3	Das Chowdhury-Schadschneider-Modell	33
3.2	ZA als Modell für Fußgängerdynamiken	34
3.2.1	Das bionische Zellularautomatenmodell	36
3.3	ZA zur Simulation von chemischen Effekten	37
3.3.1	Die Misch-Masch-Maschine	37
3.4	ZA in der Physik	41
3.4.1	Zelluläre Gittergase	41
3.5	ZA in der Biologie	44
3.5.1	Musterbildung: Aktivator-Inhibitor-Modell	44
3.5.2	Der Hyperzyklus	47
3.5.3	Der Planet Wator - Biologisches Modell eines Ökosystems	49
4	Zelluläre Automaten in der Musik	52
4.1	Eleonora Bilotta - Musical Dreams	52
4.2	Alan Dorin - Liquiprism	56
4.3	Hunt, Kirk, Orton Cellular Automata Workstation	58
4.4	Dale Millen - Cellular Automata Music	59
4.5	Eduardo Reck Miranda	61
4.5.1	Camus	61
4.5.2	Chaosynth	64
4.6	Peter Beyls	65
5	Mögliche Mappinstrategien	67
5.1	Horizontale Denkweise	67
5.2	Betrachtung der Symmetrie	68
5.3	Dynamik von Klängen	68
5.4	Der Nicht-Klang - eine Pause	69
5.5	Bewegung der Klänge	69

5.6	Klangdichte	70
5.7	Ausblick	70

Kapitel 1

Einführung

Der Theorie der zellulären Automaten (ZA) wurde vom Mathematiker Stanislaw Ulam im Jahre 1940 eingeführt. Er arbeitete damals, gemeinsam mit dem Mathematiker John von Neumann in Los Alamos am Manhattan-Projekt ¹. Von Neumann forschte damals auch im Bereich künstliche Intelligenz ². Wenn alles Leben logischen Regeln folgt, so von Neumann, muss es möglich sein, dass Maschinen sich selbst reproduzieren und somit Züge wirklichen Lebens zeigen. [Gerhardt Schuster, 1995, S. 159] Für sein Gedankenexperiment benötigte er einen Formalismus, der seine Konzept bestmöglich beschrieb. Stanislaw Ulams Theorie der ZA diente als Grundlage für das später entstandene Werk „Theory of self reproducing automata“. Der Automat war für von Neumann eine Maschine, die nach bestimmten Regeln Informationen verarbeiten kann. Im Kapitel 2.2 ist die zugrundeliegende Idee zu von Neumanns Theorie näher beschrieben.

Die Grundcharakteristika eines zellulären Automaten können nach [Gerhardt Schuster, 1995, S. 18f] folgendermaßen definiert werden:

- Seine Entwicklung findet in Raum und Zeit statt.
- Sein Raum ist eine diskrete Menge zahlreicher Zellen.
- Jede dieser Zellen hat nur eine endliche Anzahl möglicher Zustände.
- Die Zustände der Zellen verändern sich in diskreten Zeitschritten.

¹Das Manhattan-Projekt war ein von J. R. Oppenheimer geleitetes Forschungsprojekt in den USA zur Entwicklung der Atombombe während des zweiten Weltkrieges

²Das Forschungsfeld *künstlicher Intelligenz* beschäftigt sich mit Systemen, die von Menschen entwickelt wurden und dabei Züge von Lebewesen zeigen.

- Alle Zellen sind identisch und verhalten sich nach den gleichen Entwicklungsregeln.
- Die Entwicklung einer Zelle hängt nur ab von ihrem Zustand und dem ihrer zuvor definierten Nachbarzellen.

Der ZA besteht aus zahlreichen Komponenten, die miteinander nach definierten Regeln wechselwirken. Dieser funktioniert wie ein Organismus, dessen einzelnen Moleküle miteinander kommunizieren und aufeinander reagieren.

Seit dem Computerzeitalter sind ZA äußerst interessant für die Wissenschaft im Bereich komplexer Simulationsverfahren, Mustererkennung und verschiedener Rechenverfahren. Der Physiker Andrew Ilachinsky beschreibt ZA in seinem Buch *Cellular Automata: A discrete Universe* folgendermaßen[Ilachinsky, 2001, 7-9]:

- *As powerful computation engine* – CA allow very efficient parallel computational implementations to be made of lattice models in physics and thus for a detailed analysis of many concurrent dynamical processes in nature.
- *As discrete dynamical system simulators* – CA allow systematic investigation of complex phenomena by embodying any number of desirable physical properties. Reversible CA (characterized by the property that each site value has a unique predecessor neighborhood configuration), for example, can be used as laboratories for studying the relationship between microscopic rules and macroscopic behaviour – exact computability ensuring that the memory of the initial state is retained exactly for arbitrarily long period of time. . . . Suitable generalizable discrete models have thus far been constructed and studied for dendritic crystal growth, spatial patterns generated by reaction-diffusion for systems, discrete multiple soliton-structure dynamics, and turbulence in hydrodynamical systems.
- *As conceptual vehicles for studying pattern formation and complexity* - The central motivation is to abstract the general principles governing self-organizing structure formation.
- *As original models of fundamental physics* - The idea is to construct fundamentally discrete field theories to compete with existing continuous models...So the hope is to abstract a set of microphysical laws that reproduce known behaviour on the macro scale

1.1 Die Arbeit im Überblick

Im ersten Kapitels werden Grundlagen wie Zellraum, Nachbarschaft, Randbedingung und Zustandsmenge erläutert. Ebenfalls werden 1,- 2- und 3-dimensionale Automaten erläutert.

In Kapitel 2 wird ein historischer Überblick über die Entstehung und Anwendung zellulärer Automaten gegeben.

In Kapitel 3 wird anhand von konkreten Beispielen aus der Wissenschaft gezeigt, wie sich zelluläre Automaten dazu eignen, physikalische Phänomene und mathematische Modelle darzustellen.

In Kapitel 4 wird ein Überblick über bereits vorhandene Anwendung zellulärer Automaten im Bereich musikalischer Strukturgenese gegeben.

In Kapitel 5 wird ausgehend von den Beispielen in Kapitel 3 untersucht, ob daraus Mappingmodelle für die musikalische Strukturgenese abgeleitet werden können.

1.2 Definitionen

In den folgenden Abschnitten werden folgende Grundbegriffe eines ZA erläutert: Zellraum, Nachbarschaft, Randbedingungen, Zustandsmenge und Typen von ZA.

1.2.1 Zellraum

Der Zellraum eines zellularen Automaten besteht theoretisch aus unendlich vielen Zellen. Für theoretische Betrachtungen ist der unendliche Raum vorstellbar, in der angewandten Simulation jedoch muss die Zahl der Zellen begrenzt werden. Bei einem 1-dimensionalen Automaten ist die Zelle von quadratischer Form. Ein Quadrat, ein Hexagon oder ein Dreieck kann bei einem 2-dimensionalen Automaten als Grundform dienen. Bei einem 3-dimensionalen Automaten stellt ein Würfel eine Zelle dar. In der folgenden Arbeit wird immer von quadratischen Zellen ausgegangen.

Die Zellräume können folgendermaßen definiert sein:

- 1-dimensional:

$$L = ((i)|i \in \mathbb{N}, 0 \leq i < n) \quad (1.1)$$

- 2-dimensional:

$$L = ((i, j)|i, j \in \mathbb{N}, 0 \leq i < n, 0 \leq j < m) \quad (1.2)$$

- 3-dimensional:

$$L = \{(i, j, k)|i, j, k \in \mathbb{N}, 0 \leq i < n, 0 \leq j < m, 0 \leq k < p\} \quad (1.3)$$

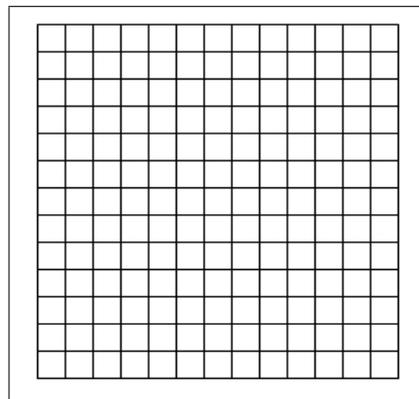


Abbildung 1.1: Bsp. für einen 2-dim. Zellraum mit $n \times m$ Zellen

1.2.2 Nachbarschaft

In einem zellularen Automaten ist die Nachbarschaft einer Zelle durch den Radius r gegeben. Wenn $r = 1$ ist, beschränkt sich die Nachbarschaft bei einem 1-dimensionalen Automaten einer Zelle auf die direkt von ihr links und rechts angrenzenden Zelle.

Bei 2-dimensionalen Automaten gibt es grundsätzlich zwei unterschiedliche Nachbarschaftstypen. In Abb. 1.2 sind diese dargestellt. Die Nachbarschaftsform, die acht Nachbarn einer Zelle berücksichtigt, nennt man Moore-Nachbarschaft, benannt nach dem Mathematiker Edward E. Moore. Weiters gibt es die von Neumann-Nachbarschaft, welche nur vier Zellen berücksichtigt, nämlich die im Norden, Osten, Süden und Westen der betrachteten Zelle. Genauso wie bei 1-dimensionalen Automaten kann auch hier der

Radius r einfach vergrößert werden.

3-dimensionale Automaten besitzen als Zelle einen Würfel. Eine Form der Nachbarschaft umfasst die sechs direkt an die Würfelflächen grenzenden Nachbarzellen. Weiters ist es auch möglich, auch die Zellen an den Würfecken zu berücksichtigen, somit ergibt sich insgesamt eine Nachbarschaft von 26 Zellen.

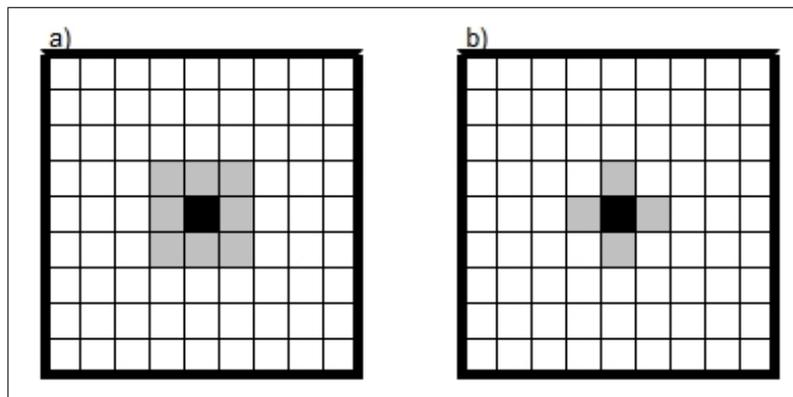


Abbildung 1.2: a) Moore-Nachbarschaft b) von Neumann-Nachbarschaft im 2-dim. Zellraum

1.2.3 Randbedingungen

Da sich der unendliche große Gitterraum nur theoretisch simulieren lässt, muss zuerst definiert werden, wie die Randzellen zu berücksichtigen sind. Bei einem 1-dimensionalen Automaten gibt es die Möglichkeit, die Zellen periodisch oder symmetrisch weiterzuführen (Abb. 1.3).

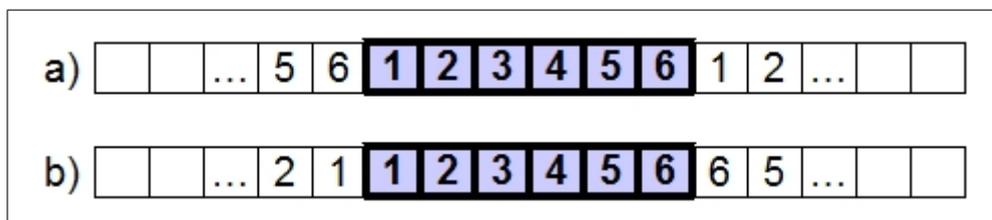


Abbildung 1.3: a) periodische b) symmetrische Randbedingung für 1-dim. Automaten

Bei 2-dimensionalen Automaten entsteht bei der periodischen Weiterführung des

Randes ein Toroid (s. Abb. 1.4). Es gibt jedoch auch die Möglichkeit, die Zellen nicht über den Rand hinauszuführen, sondern am Rand enden zu lassen.

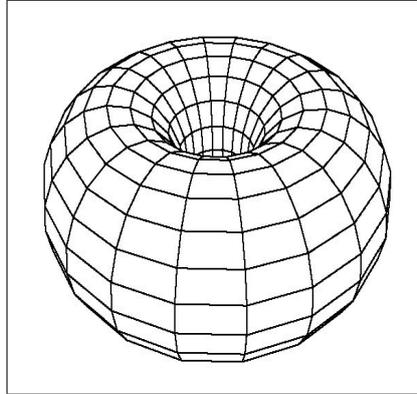


Abbildung 1.4: Toroid

1.2.4 Zustandsmenge

Die Anzahl der Zustände, die eine Zelle annimmt, kann theoretisch unendlich sein. In der Praxis zeigt sich jedoch, dass schon bei einer geringen Anzahl unterschiedlicher Zustände, viele Möglichkeiten für den Automaten entstehen und diese deswegen begrenzt und endlich sein sollten. Zustände können durch Zahlen oder Farben dargestellt werden. Wie viele Zustände für einen Automaten gewählt werden, hängt vom darzustellenden Problem ab. Schon bei einer Gittergröße von 10 x 10 Zellen und nur 2 verschiedenen Zuständen, ergeben sich hier 2^{100} Möglichkeiten für diesen Automaten.

1.3 Typen von zellularen Automaten

Die zellularen Automaten werden eingeteilt in 1-, 2- und 3- dimensionale Automaten.

1.3.1 1-dimensionaler Automat

Bei 1-dimensionalen Automaten werden Nachbarn nur in positiver und negativer x-Richtung berücksichtigt. Die Regel L für einen Automaten ist in Formel 1.4 dargestellt.

$$z_i(t+1) = L(z_{i-r}(t), z_{i-r+1}(t), \dots, z_{i+r-1}(t), z_{i+r}(t)) \quad (1.4)$$

Eine Regel L , die nur die direkt angrenzenden Zellen berücksichtigt, in diesem Fall ist der Radius $r=1$, ist in Formel 1.5 dargestellt.

$$z_i(t+1) = L(z_{i-r}(t), z_i(t), z_{i+r}(t)) \quad (1.5)$$

Zusätzlich kann auf der y -Achse die zeitliche Entwicklung des Automaten dargestellt werden (s. Abb. 1.5), wodurch bestimmte Muster erkennbar werden. Aufgrund der zeitlichen Entwicklung und der resultierenden Muster verschiedener Regeln hat der Mathematiker und Informatiker Stephen Wolfram 1-dimensionalen Automaten in vier Klassen eingeteilt:

- Klasse 1: Unabhängig von der Anfangskonfiguration erreichen alle Zellen den gleichen Endzustand.
- Klasse 2: Abhängig von der Anfangskonfiguration ist der Endzustand der Zellen variabel. Es entsteht ein einfaches, periodisches Muster.
- Klasse 3: Das Muster ist chaotisch und aperiodisch.
- Klasse 4: Komplexe Muster entwickeln sich ohne definierten Endzustand.

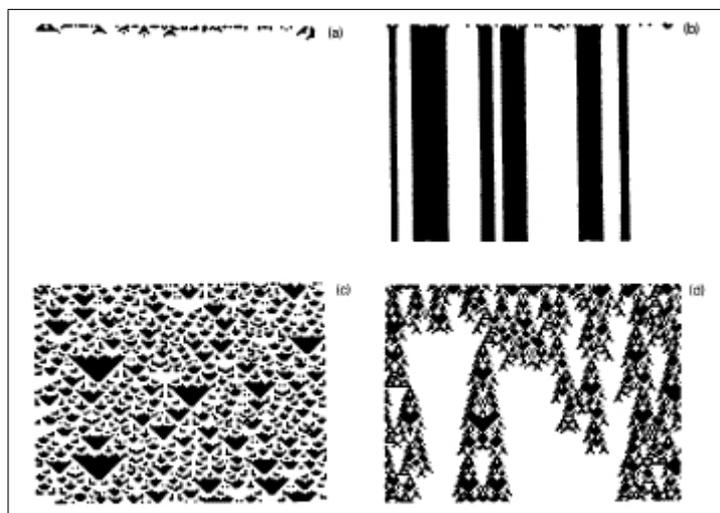


Abbildung 1.5: 4 Klassen des 1-dim Automaten

A. Ilachinsky hat noch weitere Eigenschaften der einzelnen Klassen untersucht und ist dabei zu folgenden Erkenntnissen gekommen, die in Tab. 1.1 angeführt sind.

Class	Finite Seed	Random Seed ¹	Difference Pattern ²	Predictability
1	pattern disappears	all sites attain same value	no change	completely predictable
2	evolves to fixed finite size	simple stable or periodic structures	changes only over finite region	locally predictable from local initial state
3	grows indefinitely	chaotic aperiodic behaviour	changes over expanding region	behaviour depends on ever increasing initial region
4	alternating growth and decay	complex localized structures	irregular changes	effectively unpredictable

¹ The initial configurations are defined by randomly assigned values with equal probability.

² Are space-time patterns of the difference between two evolutions of the same rule starting from two different initial states.

Tabelle 1.1: Eigenschaften der vier Klassen nach Ilachinsky

1.3.2 2-dimensionaler Automat

Einem 2-dimensionalen Automaten liegt ein $n \times m$ Gitter zugrunde. Die Nachbarschaften müssen stets definiert werden. Grundsätzlich wird hier zwischen der Moore und von Neumann-Nachbarschaft unterschieden (s. Kap. 1.2.2). Die Entwicklung der Zellen erfolgt in x - und y -Richtung. Stellt man die Entwicklung des Automaten in der Zeit als zusätzliche z -Achse dar, entsteht daraus ein sogenannter pseudo-3-dimensionaler Automat. Das bekannteste Beispiel für einen 2-dimensionalen Automaten ist Conways Spiel Game of Life (s. Kap. 2.2).

1.3.3 3-dimensionaler Automat

Der Zellraum und die Zelle eines 3-dimensionalen Automaten können aus einem Quader bestehen. In diesem Fall werden die Nachbarzellen aller drei Achsen berücksichtigt. Auch hier kann die Nachbarschaft konkret definiert und der Radius bestimmt werden. In Abb. 1.6 sind jeweils zehn Zeitschritte von vier Automaten mit unterschiedlichen Regeln und Nachbarschaftsdefinitionen wiedergegeben. Bei den ersten beiden Automaten werden nur die sechs direkt an den Würfel grenzenden Nachbarzellen berücksichtigt. Die erste Regel lautet, ein Würfel wird schwarz, wenn mindestens einer der sechs Nachbarn im vorigen Zeitschritt schwarz war. Beim zweiten Automaten wird eine Zelle nur schwarz, wenn

genau eine Nachbarzelle zuvor schwarz war. Bei den weiteren zwei Automaten werden die selben Regeln angewandt, jedoch berücksichtigt man 26 Nachbarzellen, also Zellen die direkt an eine Seite des Würfels angrenzen sowie auch an den Ecken.

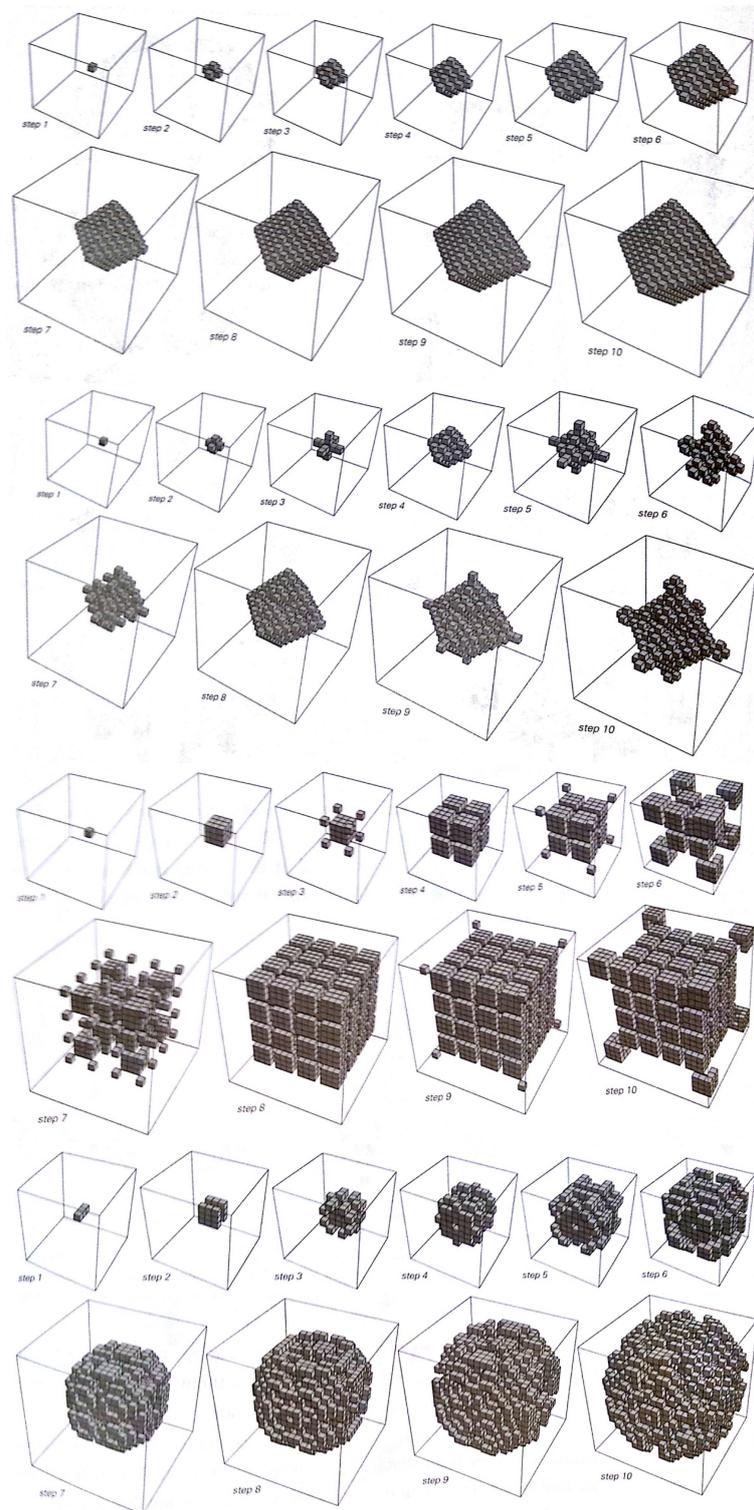


Abbildung 1.6: 3-dim. Automat

Kapitel 2

Historischer Überblick

Im folgenden Kapitel werden die wichtigsten Entwicklungen in der Geschichte des zellulären Automaten zusammengefasst. Ein kurzer historischer Überblick ist in Tab. 2.1 gegeben.

Die Physiker Wilhelm Lenz und Ernst Ising entwickelten gemeinsam Theorien zum Ferromagnetismus. Das berühmt gewordene *Ising-Model* (s. Kap. 2.1), das aufzeigt wie Ferromagnetismus mikroskopisch beschrieben werden kann, kann mit einfachen Mitteln in einen ZA übersetzt werden. Stanislaw Ulam und John von Neumann arbeiteten gemeinsam am Manhattan-Projekt in Los Alamos. Ulam führte in den 1940er Jahren das Denkmodell und den Begriff der zellulären Automaten ein. Inspiriert von der Arbeit Stanislaw Ulams, arbeitete John von Neumann an der *Theory of self reproducing automata* (s. Kap. 2.2). In diesem Gedankenexperiment beschreibt von Neumann eine Maschine, die sich selbst reproduzieren kann. Aufgrund seines Todes 1957 konnte das Werk erst von seinem Mitarbeiter Arthur Burks vollendet und 1966 publiziert werden. [Bur70] Kurz darauf folgte eine Idee von Konrad Zuse. In seiner Veröffentlichung *Rechnender Raum*

Jahr	Entwickler	Entwicklung
1922	Lenz/ Ising	Ising-Modell (Ferromagnetismus)
1940	Ulam	Einführung des Begriffes <i>Zellularer Automat</i>
1948	Neumann	Theory of self-reproducing Automata
1969	Zuse	Rechnender Raum
1970	Conway	Game of Life
2002	Wolfram	New Kind of Science

Tabelle 2.1: Historischer Überblick

[Zu69], unterteilt er den gesamten Weltraum in diskrete Zellen und versucht eine Brücke zwischen Physik und Informationstheorie zu schaffen (s. Kap. 2.3). Sehr populär wurde das Spiel *Game of Life* von John Horton Conway, welches auf einem zwei-dimensionalen zellularen Automaten beruht (s. Kap. 2.4) und nur aus drei einfachen Regeln besteht. In den 70er Jahren forschten Tomaso Toffoli und Norman Margolus auf dem Gebiet der Simulation physikalischer Vorgänge durch zelluläre Automaten. Im Jahr 2002 erschien das Buch *New Kind of Science* [Wol02] von Stephen Wolfram (s. Kap. 2.5). Seiner Meinung nach können komplexe Phänomene in der Wissenschaft mit einfachen Algorithmen, in diesem Fall zelluläre Automaten, beschrieben werden.

2.1 Das Ising-Modell

Wilhelm Lenz und sein Schüler Ernst Ising beschäftigten sich Anfang der 20er Jahre im Bereich der statistischen Physik mit dem Phänomen des Ferromagnetismus ¹. Ising stellte sich vor, ein ferromagnetischer Stoff bestehe aus vielen Elementarmagneten, die in eine bestimmte Richtung ausgerichtet sind [Gerhardt Schuster, 1995, S. 125]. Die sogenannten *Spins* beschreiben das magnetische Moment der Atome. Diese werden in einem abstrakten Modell an einem Gitter ausgerichtet und können nur zwei unterschiedliche *Zustände/Richtungen*, bezogen auf eine Achse, annehmen (*Spin Up*, *Spin Down*, $S_i = \uparrow \downarrow$). In einem eindimensionalen Modell könnte eine mögliche Konfiguration folgendermaßen aussehen:

... $\uparrow \downarrow \downarrow \downarrow \uparrow \downarrow \uparrow \uparrow \downarrow \downarrow$...

Der Zustand eines Spins ist abhängig von der Wechselwirkung mit seinen Nachbarzuständen (beschrieben durch die Kopplungskonstante J) und dem äußeren Magnetfeld B . Das ganze System wird beschrieben durch die Hamiltonfunktion, wobei μ die Magnetisierung darstellt:

$$H_{s_i} = -J \sum_{i,j} S_i S_j - \mu B \sum_i S_i \quad (2.1)$$

Bei niedriger Temperatur befindet sich das gesamte System in einem niedrigen Energiestatus in dem alle Spins in eine Richtung zeigen. Bei Erwärmung wird eine Fluktuation

¹Ein Material ist ferromagnetisch, wenn es durch ein äußeres Magnetfeld selbst magnetisiert wird bzw. selbst ein magnetisches Feld verursacht.

der Spins sichtbar. Das Ziel dieses Modells ist nicht die genaue Vorhersage der einzelnen Bewegung der Spins, sondern die Wahrscheinlichkeit des Auftretens des jeweiligen Spin.

2.1.1 Das Ising-Modell als zellularer Automat

Die Zellen eines ein - oder zweidimensionalen Gitters können zwei unterschiedliche Zustände annehmen (etwa 0 für Spin Down und 1 für Spin Up). Nur die direkt aneinandergrenzenden Zellen gelten als benachbart (in einem zweidimensionalen Automaten entspricht das der von Neumann-Nachbarschaft). Eine Zustandsänderung einer Zelle erfolgt nur dann, wenn genau zwei ihrer Nachbarn den Wert 1 besitzen. ([Gerhardt Schuster, 1995])

2.2 John von Neumann

The Theory of Self-Reproducing Automata

Der österreichisch-ungarische Mathematiker John von Neumann war Wissenschaftler an der Universität in Princeton. Er war neben Albert Einstein einer der ersten Professoren am *Institute for Advanced Study* und arbeitete in Göttingen an den Grundlagen der Quantenmechanik. An der Entwicklung der Atombombe in Los Alamos im Rahmen des Mannhattan-Projekts war neben Neumann auch der Physiker und Mathematiker Stanislaw Ulam beteiligt. Ulam entwickelte ein Prinzip zur Entwicklung komplexer rekursiver Muster. Diese geometrischen Objekte bestehen aus einzelnen Zellen, die abhängig von ihren umgebenden Zellen ihre Zustände ändern. Auch von Neumann leistete etliche Beiträge in der Mathematik, u.a. entwickelte er die Spieltheorie². Ebenfalls zu seinem Arbeitsfeld zählte die Infomatik. Nach ihm wurde die Rechnerarchitektur von Neumann-Architektur³ benannt. Von Neumann war davon überzeugt, dass das Leben und die Welt auf logischen Prinzipien beruht. Alle Organismen, auch Mensch und Tier, seien nur Maschinen die Information nach logischen Regeln verarbeiten und einen Schritt nach dem anderen ausführen [Gerhardt Schuster, 1995, S. 160]. Demnach sollte es seiner Meinung nach möglich sein, Maschinen zu bauen die wie lebendige Organismen wirken.

²Die Spieltheorie besagt, dass das eigene Handeln in Konfliktsituationen auch von Aktionen anderer abhängt und beeinflusst wird.

³Die von Neumann-Architektur beschreibt eine Single-Instruction, Single- Data-Architektur, in der ein gemeinsamer Speicher Computerbefehle und Daten verarbeitet.

Das wichtigste Kriterium für Neumann war dabei die Frage, was denn über Leben und Nicht-Leben entscheidet - für ihn war es die Fortpflanzung. Wenn sich Maschinen von selbst reproduzieren können, erst dann ist eine ausreichende Ähnlichkeit zwischen Maschinen und Lebewesen gegeben. Von Neumann stellte sich folgende die Frage:

What kind of logical organization is sufficient for an automaton to control itself in such a manner that it reproduces himself? [Burks, 1970]

Von Neumann schwebte ein sich-selbst-reproduzierender Automat vor, im Sinne einer Maschine, die nach logischen Gesetzen Information verarbeitet. Die Bestandteile dieses Automaten sind Elemente zur Speicherung und Verarbeitung von Information, ein Arm, mit dem es Teile aus dessen Umgebung nehmen und mit anderen Teilen verbinden kann und ein Speicherband, das die Bauanleitung für den nächsten Automaten enthält. In von Neumanns Vorstellung befand sich dieses Konstrukt in einem eigenen Lebensraum, welches umgeben war von unendliche vielen Bausteinen die als Grundlage für einen neuen Automaten dienten. Da dieses Gedankenexperiment sehr komplex war, gelang es von Neumann nicht, den Automaten durch einen mathematischen Formalismus vollständig zu beschreiben. Die zugrundeliegende Theorie kam von seinem Kollegen und ebenfalls Mathematiker Stanislaw Ulam. Der komplexe Lebensraum, der als Grundlage für die sich-selbst-reproduzierende Maschine dienen soll, wird durch ein großes Gitter, das aus einzelnen Zellen besteht, beschrieben. Die Zellen sind jeweils einzelne Automaten, die mit ihren benachbarten Zellen durch Informationsaustausch in Verbindung stehen und wechselwirken. Durch die gemeinsame Arbeit mit Ulam konnte von Neumann nun auf das Prinzip der zellularen Automaten zurückgreifen und seinen Automaten in einen mathematischen Formalismus übersetzen. Das Resultat war dennoch ein unheimlich kompliziertes Regelwerk mit 29 verschiedenen Zuständen und einem Gitter, welches aus mehreren hunderttausend Zellen besteht. [Von Neumann, Burks, 1966] Von Neumann verstarb im Jahr 1957 an Krebs und erst 1966 konnte sein Kollege und Freund Arthur Burks die Theorie vollenden und veröffentlichen.

2.3 Konrad Zuse - Rechnender Raum

1969 wurde das Werk *Rechnender Raum*, verfasst vom Erfinder und Bauingenieur Konrad Zuse, veröffentlicht. Er beschreibt die Automatentheorie⁴ als neuen Zweig der Wissenschaft, der sich in Zusammenarbeit mit der Informationstheorie entwickelt hat und führt den Begriff der *automatentheoretischen Denkweise*⁵ ein. In diese Richtung werden folgende Gedanken entwickelt [Zuse, 1969, S. 1f]:

1. *Eine Entwicklung und Bereitstellung von algorithmischen Verfahren, welche dem Physiker als neue Werkzeuge dienen können, seine theoretischen Erkenntnisse in praktische Ergebnisse umzuwandeln.*

2. *Es kann aber auch eine direkte Einflussnahme insbesondere automaten-theoretischer Gedankengänge auf die physikalischen Theorien selbst gedacht werden.*

Er fragt sich, ob es möglich ist, Begriffe aus der Physik wie Energie oder Wirkungsquantum durch Begriffe aus der Informationstheorie zu interpretieren, um in der Physik automatentheoretisch zu denken.

2.3.1 Der zellulare Automat im Rechnenden Raum

Als eine Spezialform eines Automaten beschreibt Zuse den ZA, der aus einem Gitter besteht, der mit den Nachbarpunkten durch Informationsaustausch in Verbindung steht [Zuse, 1969, S. 5]. Durch die Idee der Gitterstruktur des Raumes soll der Kosmos durch diskrete Automaten beschrieben werden. Um den Bezug zur Physik herzustellen, führt er eine Art Elementarteilchen in seiner Theorie ein und nennt es Digitalteilchen.

Digitalteilchen lassen sich als Störung eines Normalzustandes eines zellularen Automaten auffassen. Diese Störung hat ein bestimmtes Muster, welches einer periodischen Änderung unterworfen ist. Im Sinne der Automatentheorie geht jeder folgende Zustand aus dem vorhergehenden hervor, jedoch kann sich das gesamte Muster dabei bewegen.

⁴Der Begriff *Automat* kommt vom griechischen Wort *automatos*, sich selbst geschehend. Dementsprechend bezeichnet die *Automatentheorie* maschinelle Vorgänge, die ohne äußere Einwirkung passieren und stellt eine Reihe von allgemeinen Gesetzen über ihre Arbeitsweise auf. [Zuse, 1969]

⁵Unter automatentheoretischer Denkweise wird eine Betrachtungsweise verstanden, bei der technische, mathematische oder physikalische Modelle unter dem Gesichtspunkt eines Ablaufs von Zuständen gesehen werden, die gesetzmäßig aufeinander folgen. [Zuse, 1969]

Wir haben es gewissermaßen mit 'fließenden Zuständen' zu tun. In diesem Sinne kann man Digitalteilchen auch als sich selbst reproduzierende Systeme auffassen. Ein gegebenes Muster erzeugt sich selbst in einem Nachbargebiet des zellularen Automaten. [Zuse, 1969, S.50].

Zuse ist sich bewusst, dass die Digitalteilchen noch viel zu einfach sind, um in engere Beziehung zu physikalischen Vorgängen gesetzt zu werden. Physikalische Gesetze wie Energie - und Impulserhaltungssatz können mit seinem Modell nicht dargestellt werden. Er entwickelt weiterführende Gedanken wie die Einführung verschiedener Teilchentypen (analog zur Physik wie Photonen, Elektronen etc.) und die unterschiedliche Fortpflanzungsrichtung der einzelnen Digitalteilchen.

Wenn auch die vorhergehende Betrachtung noch nicht zu handgreiflichen Lösungen führen, so dürfte doch gezeigt sein, dass der vorgeschlagene Weg einige neue Perspektiven eröffnet... [Zuse, 1969, S. 68]

Als Ausblick versucht Zuse noch eine mögliche Gegenüberstellung verschiedener Auffassungen der Physik und seiner Ideen (Tab. 2.2).

2.4 John Conway - Game of Life

Der ZA *Game of Life*, im folgenden GoL genannt, wurde 1968 vom englischen Mathematiker John Horton Conway an der mathematischen Fakultät der Universität Cambridge entwickelt. Grundsätzlich gibt es in GoL zwei Arten von Zellen: lebendige und tote. Es gibt lebendige Zellen, die einerseits um das Überleben kämpfen, aber auch an Vereinsamung sterben können. Diese Idee hat Conway in ein Spiel mit nur wenigen Regeln zusammengefasst.

Der zweidimensionale Zellraum besteht aus einem rechteckigen $m \times n$ Gitter. Die einzelnen Gitterzellen sind entweder *lebendig*, symbolisiert durch eine 1 oder eine schwarze Zelle; oder *tot*, dargestellt durch eine 0 oder eine weiße Zelle, d. h. sie können zwei verschiedene Zustände annehmen. Durch die Moore-Nachbarschaft werden die acht direkt angrenzenden Zellen bei einer Zustandsänderung berücksichtigt. Die Änderung dieser Zustände in diskreten Zeitschritten basiert auf folgenden drei Regeln, die Conway *Ge-*

Klassische Physik	Quantenphysik	Rechnender Raum
Punktmechanik	Wellenmechanik	Automatentheorie, Schaltalgebra
Korpuskel	Welle-Korpuskel	Schaltzustand, Digitalteilchen
analog	hybrid	digital
Analysis	Differentialgleichungen	Differentialgleichungen und log. Operationen
kontinuierliche Größen	gequantelte Größen	diskrete Werte
keine Grenzwerte	Lichtgeschwindigkeit	Minimal- Maximalwerte aller Größen
unendlich genau	Unbestimmtheitsrelation	Begrenzte Rechengenauigkeit
Kausalität in beide Zeitrichtungen	nur statistische Kausalität, Auflösung in Wahrscheinlichkeit	Kausalität nur in positive Zeitrichtung, Einführung von Wahrscheinlichkeitstermen möglich, aber nicht nötig
	Klassische Mechanik wird statistisch angenähert	Wahrscheinlichkeitsgesetze der Quantenphysik durch determinierte Raumstruktur erklärbar?
	Urformeln ¹	Urschaltung ²

¹ Der Begriff Urformel bzw. Weltformel kommt aus der Mathematik und theoretischen Physik. Sie ist eine hypothetische Theorie das alle grundlegende Wechselwirkungen in der Natur erklären würde

² Analog zur Weltformel dachte sich Zuse auch eine alles erklärende Urschaltung.

Tabelle 2.2: Gegenüberstellung versch. möglicher Auffassungen [Zuse, 1969, s. 68]

Zellraum	2-dimensional, m x n Gitter
Nachbarschaft	Moore
Zustandsmenge	(0,1) 0=tote Zelle, 1=lebendige Zelle
Anfangszustand	beliebig
Randbedingungen	beliebig
Zustandsentwicklung	$z_{t+1} = \begin{cases} 1 : \sum_{(k,l) \in \mathbb{N}_{ij}} z_{k,l}(t) = 3 \\ 1 : \sum_{(k,l) \in \mathbb{N}_{ij}} z_{k,l}(t) = 4 \text{ und } z_{i,j}(t) = 1 \\ 0 : \text{sonst} \end{cases}$

Tabelle 2.3: Zusammenfassung GoL [Gerhardt Schuster, 1995, S.35]

bert, Tod und Überleben nennt:

- *Geburt*: Ersetzen einer toten Zelle mit einer lebendigen wenn genau drei ihrer Nachbarn lebendig sind.
- *Tod*: Ersetzen einer lebendigen Zelle mit einer toten wenn genau a) nur eine lebendige Zelle in der Nachbarschaft vorhanden ist (stirbt durch Isolation), oder b) mehr als drei lebendige Zellen in der Nachbarschaft vorhanden sind (stirbt aufgrund Überbevölkerung)
- *Überleben*: Beibehalten der lebendigen Zelle wenn genau zwei oder drei lebendige Zellen in der Nachbarschaft vorhanden sind.

2.4.1 Organismen in GoL

Eine Ansammlung von lebendigen Zellen kann als ein Organismus von GoL angesehen werden. Besteht ein Organismus aus nur zwei lebenden Zellen, gibt es keine Überlebensmöglichkeit und er stirbt zum nächsten Zeitpunkt aus. Das Minimum an aktiven Zellen für die Garantie zur Erhaltung von Leben besteht aus zwei lebenden Nachbarn. Für die Generierung von neuem Leben werden mindestens drei lebende Zellen in der Nachbarschaft benötigt. Der kleinste lebensfähige Organismus ist in Abb. 2.1 dargestellt. [Gerhardt Schuster, 1995, S. 37] Bild a) zeigt eine simple Anordnung von drei benachbarten Zellen, die den Anschein erwecken, sich bei jedem Zeitschritt um 90° zu drehen

und somit ein oszillierendes Muster ergeben. Conway und seine Kollegen nannten diesen Organismus *Blinker* [Gerhardt Schuster, 1995]. Bild b) zeigt einen Block, der unverändert existiert, solange dieser nicht von einer unmittelbaren Nachbarzelle *gestört* wird. Das Überleben des Organismus ist gesichert, da jede Zelle drei aktive Nachbarn besitzt, jedoch besteht keine Möglichkeit zur Fortpflanzung.

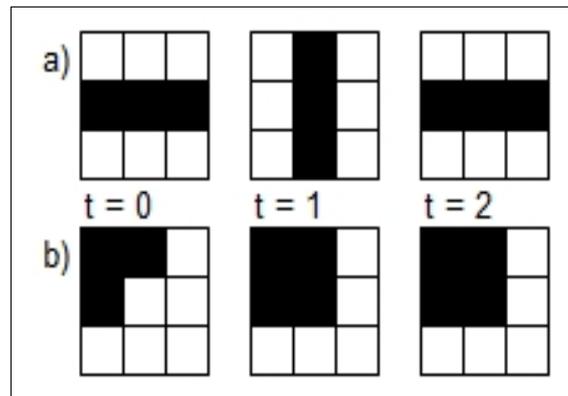


Abbildung 2.1: a) Blinker und b) Block in GoL

Somit gibt es bei diesem Spiel einerseits Organismen, die oszillierende Muster mit einer bestimmten Periodenlänge ergeben, aber auch statische Gebilde, die in einem Endzustand verharren, solange sie keine lebendige Zelle in ihrer Nachbarschaft *stört*. In Abb. 2.2 ist ein Beispiel, wie aus einer simplen Startkonfiguration von fünf lebendigen Zellen, es innerhalb von acht Generationen zu einem oszillierenden Muster kommt, welches aus dem so genannten *Blinker* besteht.

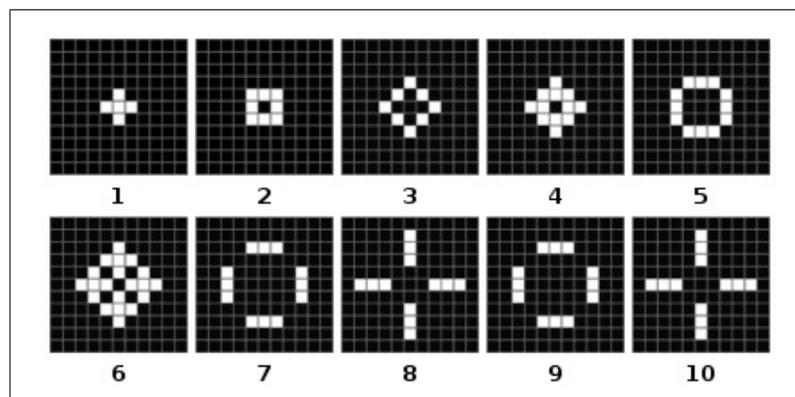


Abbildung 2.2: Entwicklung einer simplen Startkonfiguration

Diese oszillierenden Organismen treten in einer Vielzahl von unterschiedlichen Konfigurationen auf. Sobald sie einen Endzustand erreicht haben, wird dieser nicht mehr

verlassen und die Position bleibt innerhalb einer bestimmten Periodenlänge die Gleiche. Es gibt bei diesem Spiel aber auch noch andere Organismen, die nicht statisch an einen Fleck verweilen sobald sie ihren oszillierenden Endzustand erreicht haben, sondern den Anschein erwecken, diagonal über die Gitterfläche zu wandern. Einer der bekanntesten Formen ist die sogenannte *glider-gun* (s. Abb.2.3). Alle 30 Schritte wird ein *glider*, bestehend aus fünf lebendigen Zellen, aus der *glider-gun* geschossen, welche den Anschein erwecken diagonal oszillierend über den Bildschirm zu wandern. Auf diese Art und Weise gelingt es statische Organismen wieder zu verändern und völlig neue Wesen zu schaffen. Ebenso kann es aber auch passieren, wenn zwei glider zusammenstoßen, dass sie sich dabei vollkommen auslöschen (s. Abb.2.4).

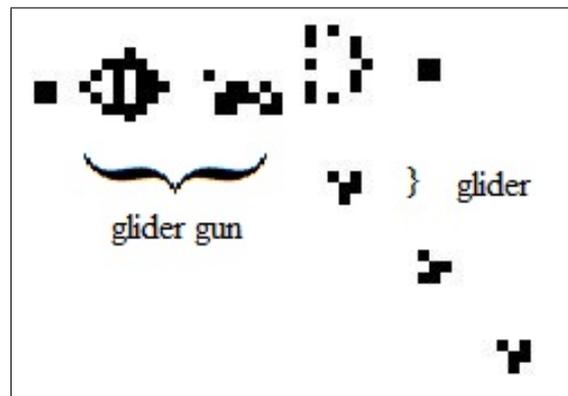


Abbildung 2.3: Glider Gun

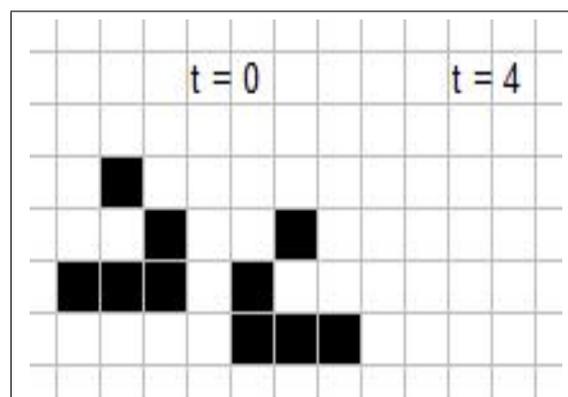


Abbildung 2.4: Auslöschung

Eines weiterer interessanter Organismus in GoL ist das *r-Pentomino*. Der Organismus, bestehend aus fünf lebendigen Zellen, sieht aus wie der Buchstabe *r* (s. Abb.2.5)

und schafft sozusagen neues Leben. In Abb. 2.5 sind die Zeitschritte $t=0$, $t=75$ und $t=150$ dargestellt. Es wurden eine Vielzahl von lebendigen Zellen geschaffen, sogar *glider* befinden sich unter ihnen. Kein Wunder, dass das r-Pentomino zu den fruchtbarsten Organismen im Spiel gehören. [Gerhardt Schuster, 1995, S. 41]

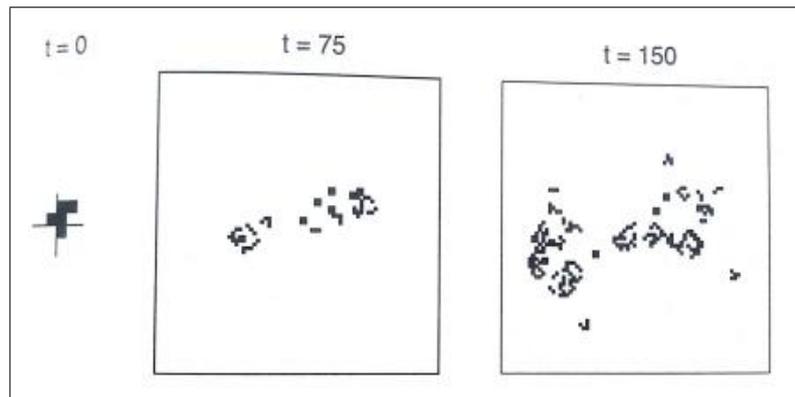


Abbildung 2.5: r-Pentomino

Im Gegensatz dazu gibt es aber auch Wesen, die lebende Zellen regelrecht auffressen. In Abb. 2.6 sind solche Organismen dargestellt. Innerhalb weniger Zeitschritte frisst diese Ansammlung von sieben lebendigen Zellen ihre Nachbarn in der Umgebung auf. Der Organismus selbst trägt dabei keine Schäden davon und kann weiter auf Jagd gehen.

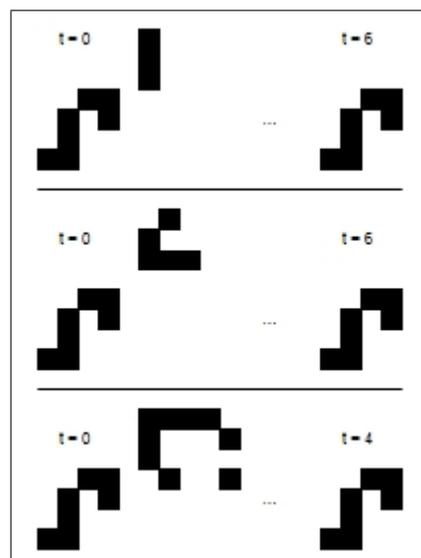


Abbildung 2.6: Fresser

Conway selbst sagt über sein Spiel:

It's probable, given a large enough Life space, initially in a random state, that after a long time, intelligent self-reproducing animals will emerge and populate some parts of the space.[Ilachinsky, 2001, S. 133]

2.5 Stephen Wolfram - A New Kind of Science

Stephen Wolfram, ein Physiker aus Großbritannien, verfasste das Werk *The New Kind of Science* im Jahre 2002. In seinem Buch beschreibt er, welche Möglichkeiten ZA für die Wissenschaft bieten. Für ihn stellen ZA eine einzigartige Möglichkeit dar, mit simplen Modellen, komplexe Problemstellungen aus der Physik, Biologie, den Geisteswissenschaften etc. zu beschreiben. Nicht immer muss ein komplex wirkendes System auch komplex sein:

Whenever a phenomenon is encountered that seems complex it is taken almost for granted that the phenomenon must be the result of some underlying mechanism that is itself complex. But my discovery that simple programs can produce great complexity makes it clear that this is not in fact correct.[Wolfram, 2002, S. 4]

Wolfram hat ZA eingehend untersucht und dabei entdeckte er Anfang der 80er Jahre, dass diese, unabhängig von der Anzahl der unterschiedlichen Zustände, aufgrund ihrer Fortschreitng in vier verschiedene Klassen eingeteilt werden können (s. Tab. 2.4). Es gibt einige Zellulare Automaten, deren Eigenschaften auch in zwei Klassen gefunden werden können und nicht direkt klassifizierbar sind, diese sind jedoch in der Unterzahl. Um grundsätzliche Aussagen über eine bestimmte Regel machen zu können, bzw. die Zuteilung zu einer Klasse zu schaffen, ist es nötig, die Entwicklung eines Automaten mit unterschiedlichen Anfangskonfigurationen zu untersuchen. Nur so ist es möglich, einen bestimmten Automaten in eine Klasse einzuordnen.

Wolfram entdeckte auch, dass es auch schon mit 1-dimensionalen Automaten möglich ist, komplexe Phänomene zu simulieren. Bei diesen kommt sozusagen die Zeit als zweite Dimension hinzu und Struktur wird sichtbar, wenn die räumlichen Zustände durch Aneinanderreihung in einem zeitlichen Verlauf dargestellt werden (s. Abb. 2.7).

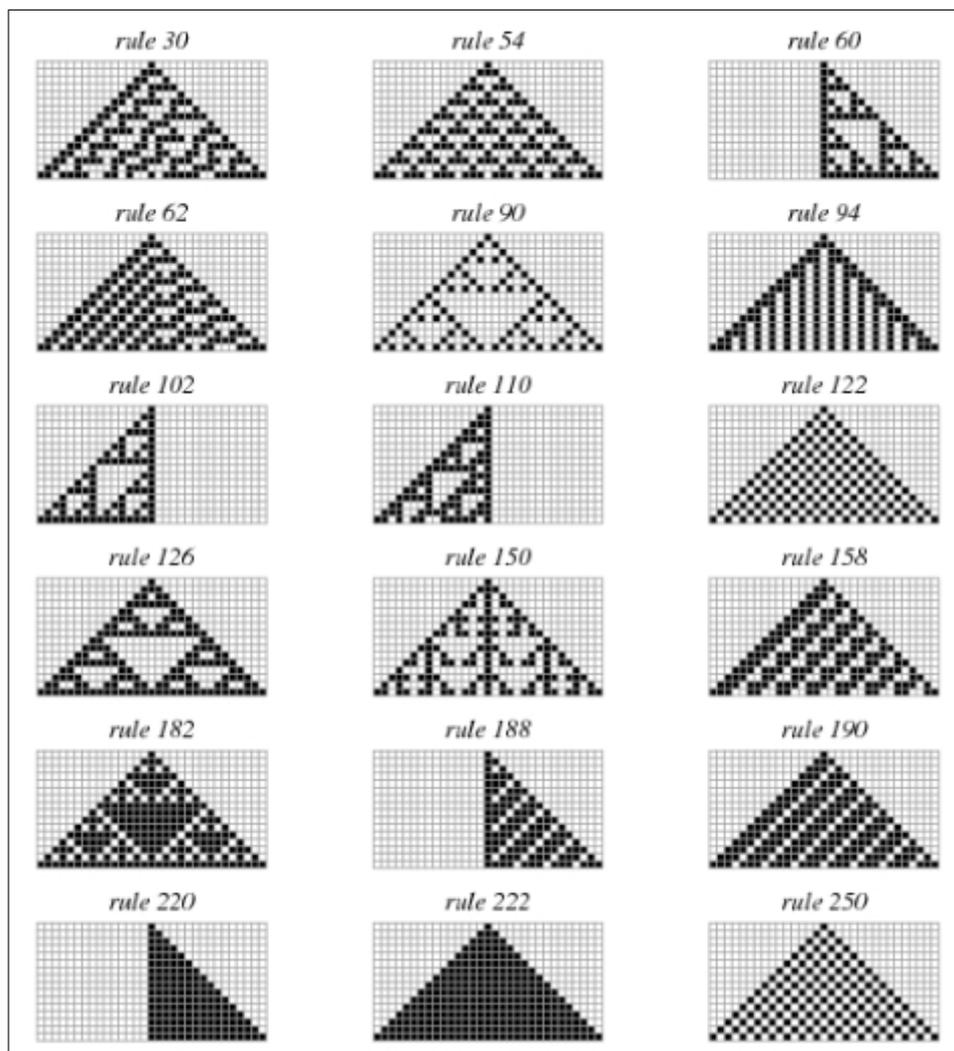


Abbildung 2.7: 1-dim. Automaten [Wolfram, 2002]

Class	Evolution
1	The behavior is very simple, and almost all initial configurations lead to exactly the same uniform final state.
2	There are many different possible final states, but all of them consist just of a certain set of simple structures that either remain the same forever or repeat every few steps.
3	The behaviour is more complicated, and seems in many respects random, although triangles and other small-scale structures are essentially always at some level seen.
4	Involves a mixture of order and randomness: localized structures are produced which in their own are fairly simple, but these structures move around and interact with each other in very complicated ways.

Tabelle 2.4: Einteilung in vier Klassen [Wolfram, 2002]

2.5.1 Klasse 1

In dieser Klasse erreichen alle Automaten, bis auf einige wenige Ausnahmen, unabhängig von der Anfangskonfiguration den gleichen Zustand. Das Muster stellt sich nach nur wenigen Zeitschritten ein und bleibt erhalten. Wird ein solcher Automat auf dem Bildschirm simuliert, scheint es, als ob es keine Entwicklung gibt. Jegliche Anfangsinformation wird sozusagen ausgelöscht und die Zellen gehen in einen uniformen Zustand über.

2.5.2 Klasse 2

Hier entstehen abhängig von der Anfangskonfiguration, schon nach wenigen Zyklen einfache Muster die sich periodisch wiederholen. Typisch für diese Klasse sind Längsstreifen mit unterschiedlichen Blockgrößen.

2.5.3 Klasse 3

Die Automaten der Klasse 3 scheinen auf den ersten Blick durchaus chaotisch, jedoch sind hier ebenfalls Muster zu erkennen, die sich mit der Zeit stets weiterentwickeln. Typisch sind hier Dreiecksbildungen verschiedener Größen die räumlich ungeordnet verteilt liegen. In diesem Fall treten keine wiederkehrenden Muster mit bestimmter Periodenlänge auf, was den Anschein von Chaos vermuten lässt.

2.5.4 Klasse 4

Die Zuordnung eines Automaten zur vierten Klasse fällt am schwierigsten aus. Hier entstehen keine charakteristischen Muster. Die Entwicklung hängt von der Anfangskonfiguration ab und kann völlig unterschiedlich sein. Während einige Automaten Strukturen wie die *glider-gun* bilden, besitzen andere wiederum mehrere periodisch wiederkehrende Muster.

2.5.5 Häufigkeit der Klassen im 1-dimensionalen Raum

Wolfram untersuchte für die Regeln im 1-dimensionalen Raum die Häufigkeiten der Klassen in Abhängigkeit von der Anzahl ihrer Zustände k und ihrem Nachbarschaftsradius r und stieß dabei auf folgendes Ergebnis:

Es ist zu beobachten, dass Klasse 3 am Häufigsten vorkommt und Klasse 4 am Seltensten. Aufgrund dieser Erkenntnis untersuchten Wolfram und seine Mitarbeiter auch 2-dimensionale Automaten. Dabei stellte sich heraus, dass diese sich ähnlich verhielten. [Wol02]

2.5.6 Der Langton-Parameter

Der Informatiker Christopher Langton ist Komplexitätsforscher am Santa Fe Institute in Los Alamos. Er wurde in den 80er Jahren durch die Einführung des Begriffs *Artificial Life* berühmt und experimentiert seit dem mit künstlicher Intelligenz. Langton war fasziniert von John von Neumann und Stephen Wolfram. ZA stellten für ihn eine Möglichkeit dar um künstliches Leben zu produzieren. Nachdem er sich ausgiebig mit Wolfram befasst hatte, stellte er sich eine Frage:

Wo steckt das Prinzip, das dafür sorgt, welche Regel eines Automaten in welche Klasse einmündet? [Gerhardt Schuster, 1995, S. 94]

Nach zahlreichen Experimenten fand er dann doch schließlich eine Lösung, nämlich den Langton-Parameter λ . Dieser beschreibt die Wahrscheinlichkeit, mit der eine Zelle im nächsten Zeitschritt den Wert 1 hat. Bei einem Automaten mit dem Wert $\lambda=0$ sterben alle Zellen in der folgenden Generation aus und es gibt somit keine Entwicklung. In Abb. 2.8 ist die Komplexität eines Automaten in Abhängigkeit des Lambda-Wertes angegeben. Bei einem Wert um $\lambda=0,3$ entstehen Automaten der Klasse 4.

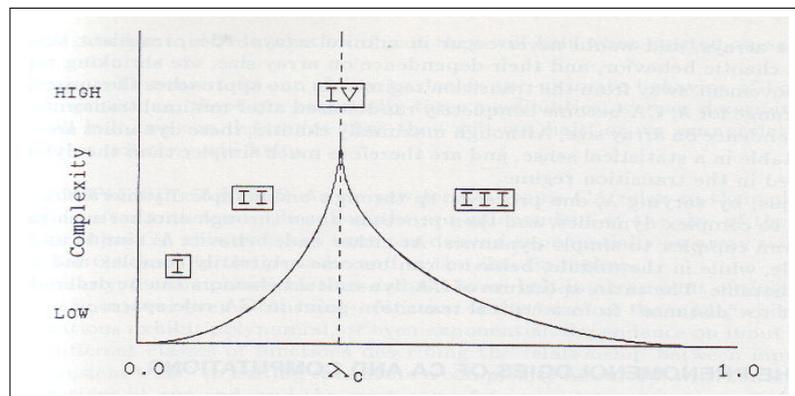


Abbildung 2.8: Komplexität in Abhängigkeit des Langton-Parameters

Kapitel 3

Zelluläre Automaten in der Wissenschaft

Im folgenden Kapitel werden einige Anwendungsbeispiele aus der Physik, dem Verkehr, der Chemie und der Mathematik näher erläutert. ZA dienen dabei als Simulationstool für Verkehrsmodelle, als Modell zur Musterbildung, zur Modellierung chemischer Reaktionen oder die Ausbreitung von Gasen. Es sei hier angemerkt, dass die folgenden Beispiele nur einen kleinen Bereich von ZA in der Wissenschaft darstellen. Jedoch wurden diese in Hinblick auf ihre Möglichkeit zur Anwendbarkeit in der musikalischen Strukturgenese ausgewählt. Der Zusammenhang zwischen den Modellen aus der Wissenschaft und der Anwendbarkeit in der Musik wird abschließend in Kapitel 5 erläutert.

3.1 ZA in der Verkehrsmodellierung

3.1.1 Das Nagel-Schreckenberg-Modell

Das Nagel-Schreckenberg-Modell (NaSCH-Modell) wurde im Jahr 1992 von den Physikern Kai Nagel und Michael Schreckenberg entwickelt. Das Modell dient zur Simulation des Straßenverkehrs, wobei einerseits der Verkehrsfluss beschrieben wird und andererseits das Phänomen der Staubbildung gelöst wird.

Das NaSch-Modell besteht aus einem 1-dimensionalen Automaten, entweder mit offenen oder periodischen Randbedingungen. Jedes Fahrzeug besitzt eine bestimmte Geschwindigkeit, die durch einen diskreten Zustand beschrieben wird und zwischen 0 und

v_{max} liegt. Insgesamt gibt es vier Regeln, die gleichzeitig auf jedes Auto angewendet werden:

- *Beschleunigung*: Ist $v_n < v_{max}$, wird die Geschwindigkeit um einen Zustand erhöht. Dies stellt das Phänomen dar, dass in den meisten Fällen die Höchstgeschwindigkeit angestrebt wird.
- *Abbremsen durch andere Autos*: Anhängig vom Abstand zum nächsten Auto, wird die Geschwindigkeit verringert, wenn der Abstand kleiner ist als die Geschwindigkeit des Autos. Diese Regel dient der Vermeidung von Unfällen.
- *Trödeln*: Wenn das Auto nicht steht, wird die Geschwindigkeit zufällig mit der Wahrscheinlichkeit p verringert. Hier wird das schwankende Fahrverhalten berücksichtigt.
- *Fahren*: Das Auto wird mit der neuen Geschwindigkeit nach rechts bewegt.

$$x_n(t+1) = x_n(t) + v_n(t+1) \quad (3.1)$$

In Abb. 3.3 ist Modell für eine Dichte von $\rho = 0,03$ dargestellt. Gut zu erkennen ist die Beschleunigung bis zu einer Geschwindigkeit von Zustand vier oder sogar fünf, bis dann ein Abbremsen folgt und es zur Staubildung kommt mit der Geschwindigkeit null. Bemerkenswert ist auch, dass der Stau immer mehr nach *links* wandert. Danach beginnen die Autos wieder zu beschleunigen. Je nach Abhängigkeit von Größe des Gitters und Dichte der Autos, kommt es zur Staubildung.

3.1.2 Das Behime-Middleton-Levine-Modell

Das Behime-Middleton-Levine-Modell (BML-Modell) wurde ebenfalls im Jahr 1992 von den gleichnamigen Physikern entwickelt. Dem Modell liegt ein quadratisches Gitter mit periodischen Rangbedingungen zugrunde. In diesem 2-dimensionalen Verkehrsmodell gibt es zwei unterschiedliche Zustände die eine Zelle annehmen kann, symbolisiert durch die Farben rot und blau. Eine blaue Zelle stellt ein nach Norden fahrendes Auto dar, eine rote Zelle ein nach Osten fahrendes Auto. Die einzige Bedingung, dass ein Auto sich vorwärts (entweder nach Norden oder nach Osten) bewegt, ist eine leere Zelle vor dem Auto. Bei ungeraden Zeitschritten dürfen sich alle blauen Autos Richtung Norden bewegen, bei geraden Zeitschritten alle roten Autos nach Osten. Die Anfangskonfiguration kann beliebig gewählt werden. Ab einer bestimmten Dichte von Autos (ca. 33%), kann

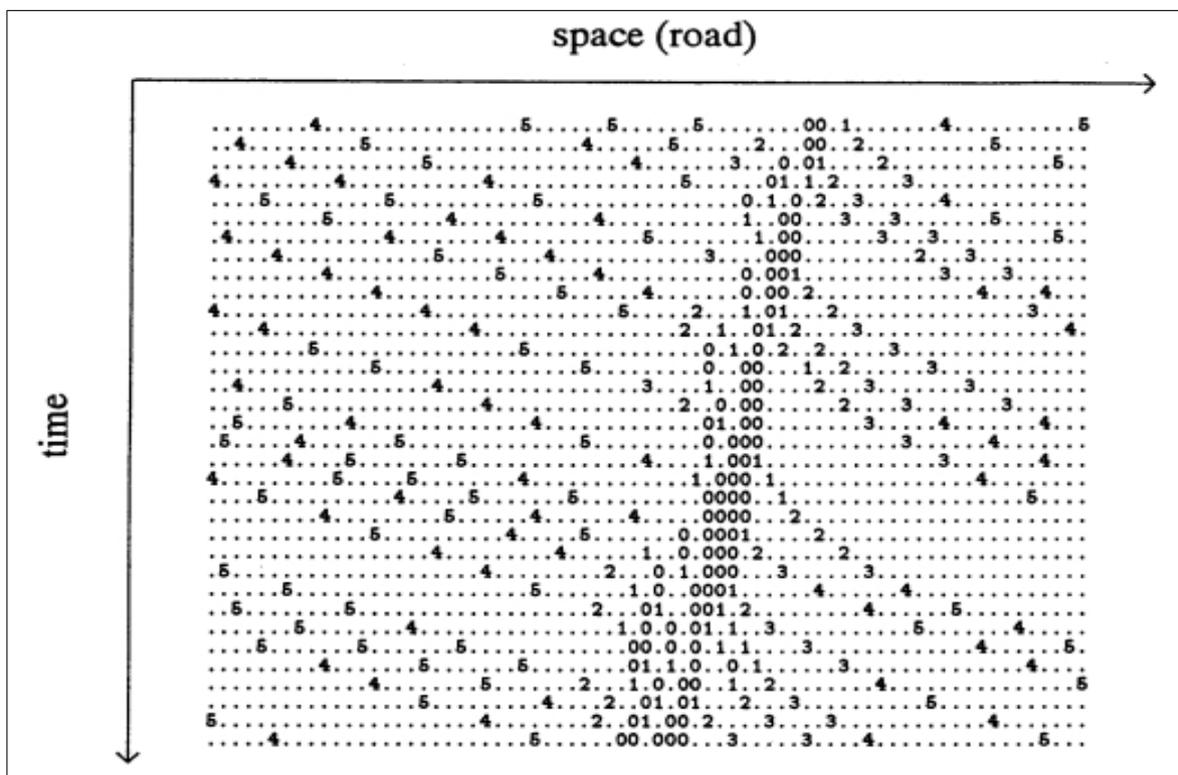


Abbildung 3.1: Verkehrssimulation mit Staubildung

es zur Staubildung kommen. Davor organisiert sich das System selbst und es entsteht ein geometrisches Muster (siehe Abb. 3.2).

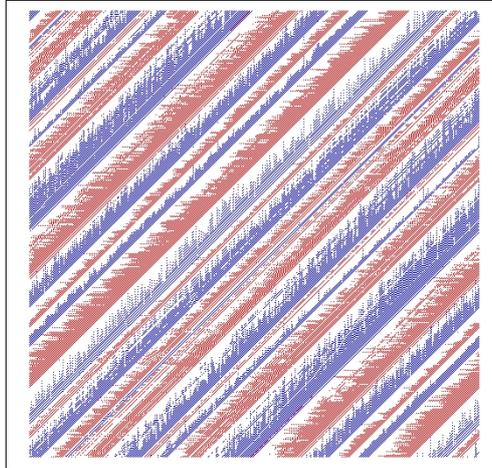


Abbildung 3.2: BML-Modell ohne Staubildung

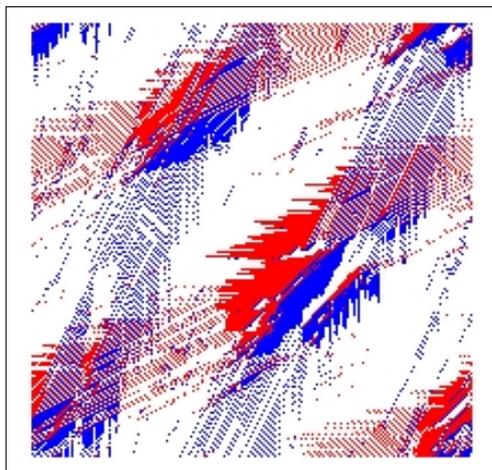


Abbildung 3.3: BML-Modell mit Staubildung

3.1.3 Das Chowdhury-Schadschneider-Modell

Das Chowdhury-Schadschneider-Modell (CS-Modell) zur Simulation der Straßenverkehrsdynamik wurde von den Physikern Debashish Chowdhury und Andreas Schadschneider entwickelt. Es geht sowohl auf die Wechselwirkung der Fahrzeuge untereinander ein, als auch auf die Wechselwirkung der Autos mit den Ampeln. Wie im BML-Modell, können sich die Fahrzeuge (aktive Zellen) nur von Süden nach Norden oder

von Westen nach Osten bewegen. Die Zustandsregeln sind an das Nasch-Modell angelehnt, mit der Ausnahme dass beim Abbremsen der Autos auch noch die rote Ampel mit berücksichtigt wird (s. Formel 3.2 bis 3.7).

- Regel 1 - Beschleunigen:

$$v_n \rightarrow \min(v_n + 1, v_{max}) \quad (3.2)$$

- Regel 2 - Bremsen durch Autos oder Ampeln:

Fall A - Ampel ist rot

$$v_n \rightarrow \min(v_n, d_n, s_n) \quad (3.3)$$

Fall B - Ampel ist grün und wird im nächsten Zeitschritt rot

$$v_n \rightarrow \min(v_n, d_n, s_n) \quad (3.4)$$

Fall C - Ampel ist und bleibt grün

$$v_n \rightarrow \min(v_n, d_n) \quad (3.5)$$

- Regel 3 - Trödeln:

$$v_n \xrightarrow{p} (v_n - 1, 0) \quad (3.6)$$

- Regel 4 - Fahren:

$$x_n \rightarrow x_n + v_n \quad (3.7)$$

Abhängig von der Größe des Modells und der Dichte der auftretenden Autos, kann ein sogenannter *Gridlock* entstehen (s. Abb. 3.5). In diesem Fall ist kein weiterkommen mehr möglich, da alle Kreuzungen blockiert werden. Zu beobachten ist, dass bei langen Ampelphasen die Geschwindigkeit aller Autos auf Null zurückgeht, während bei kurzen Ampelphasen die Durchschnittsgeschwindigkeit annähernd gleich bleibt.

3.2 ZA als Modell für Fußgängerdynamiken

Wenn es zur Modellierung von Fußgängerdynamiken kommt, handelt es sich immer um ein zweidimensionales Modell. Folgende Phänomene müssen dabei berücksichtigt werden [Schadschneider, 2004]:

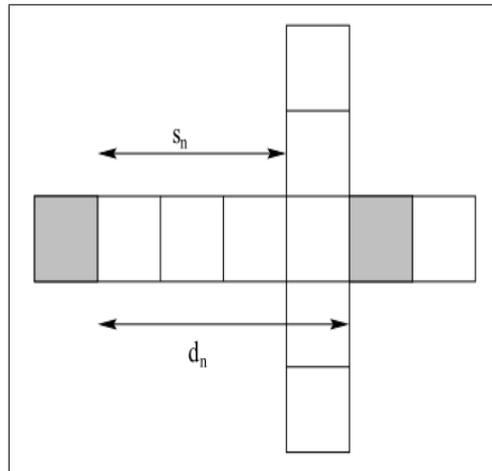


Abbildung 3.4: CS-Modell: s (Lücke zur nächsten Ampel), d (Lücke zum nächsten Fahrzeug)

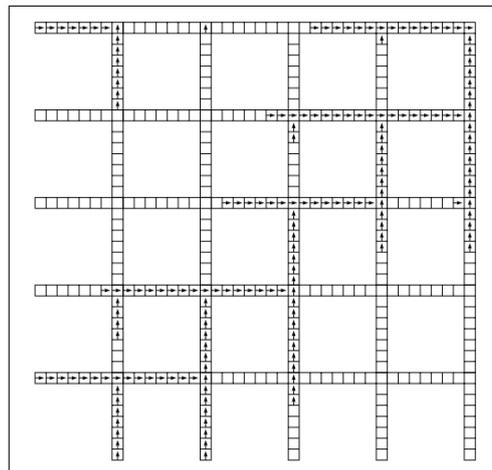


Abbildung 3.5: CS-Modell mit Gridlock

- *Staubildung*: Ein Stau kann sich bilden, wenn es zu einer Engstelle kommt (z.B. bei einem Ausgang, s. Abb. 3.6).
- *Spurbildung*: Bei Fußgängern, die sich in entgegengesetzten Richtungen bewegen, können sich selbstorganisierte Spuren (Reihen mit bestimmter Dynamik) bilden.
- *Oszillation*: Bei Engstellen und Gegenverkehr kann es dazu kommen, dass es für eine Richtung kurzzeitig zu einem Stau kommt.
- *Flussmuster*: Um ein reibungsloses Fortkommen zu gewährleisten, entsteht an Kreuzungen mit einem hohen Fußgängeraufkommen oft ein Kreisverkehr (s. Abb. 3.7).
- *Trampelpfade*: Diese entstehen durch Abkürzungen z.B. in Parkanlagen. Durch die Wegersparnis entstehen Pfade durch Grünflächen.
- *Panikeffekte*: Durch den Fluchtinstinkt bedingte Beschleunigung der einzelnen Person kommt es zu einer Verlangsamung der gesamten Menschenmenge.

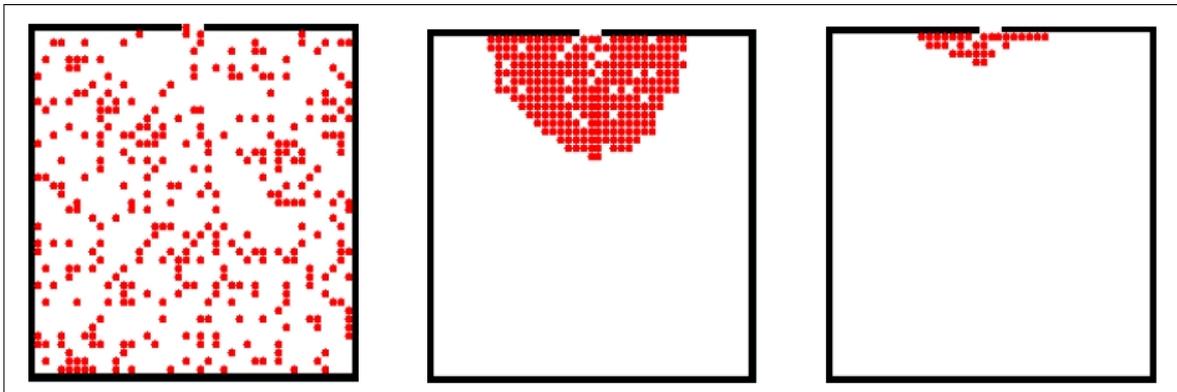


Abbildung 3.6: Phänomen der Staubildung

Im folgenden Abschnitt wird das bionische Zellularautomatenmodell näher erläutert. Dieses Modell zur Simulation von Fußgängerdynamiken stellt einen Versuch dar, die zuvor genannten Phänomene zu implementieren.

3.2.1 Das bionische Zellularautomatenmodell

Dieses Modell zur Simulation von Fußgängerdynamik wurde von den Wissenschaftlern Carsten Burstedde, Kai Klauck, Andreas Schadschneider und Johannes Zittartz im

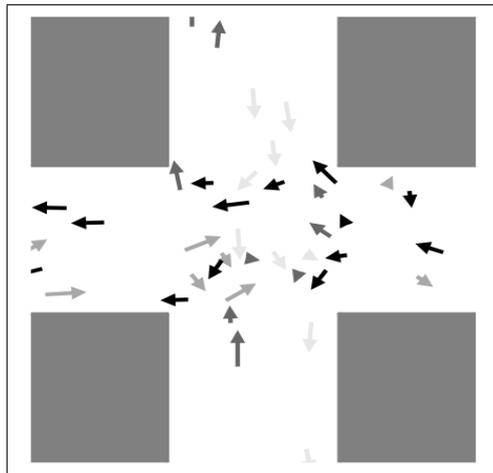


Abbildung 3.7: Entstehung eines Kreisverkehrs

Jahre 2001 entwickelt. Das Modell besteht aus zwei 2-dimensionalen Gittern. Beim ersten Gitter simulieren aktive Zellen Fußgänger. Das zweite Gitter stellt ein sogenanntes *Bodenfeld* dar (s. Abb. 3.8), [Schadschneider, 2004] nennt es auch *Bosonenfeld*¹, da sich in einer Zelle mehrere Teilchen befinden können (realisiert durch unterschiedliche Zustände). Unterschieden wird zwischen einem dynamischen und einem statischen Bodenfeld. Mit dem statischen Bodenfeld lassen sich Geometrieeffekte sehr gut simulieren. Abb. 3.8 zeigt, wie die Stärke des Feldes in Richtung Türe zunimmt. Das bewirkt den Effekt, dass sich alle Fußgänger in Richtung Türe bewegen. Beim dynamischen Bodenfeld wird das Feld durch die Bewegung der Fußgänger modifiziert. Bewegt sich dieser zu einer Nachbarzelle, wird die Ausgangszelle um ein Boson erhöht, was das zuvor beschriebene Phänomen der Spurbildung zufolge hat. [Schadschneider, 2004]. Ein weiteres Element bildet die Präferenzmatrix (s. Abb. 3.9). Diese enthält Felder mit unterschiedlichen Wahrscheinlichkeiten, die sich auf die Wunschrichtung der Fußgänger bezieht.

3.3 ZA zur Simulation von chemischen Effekten

3.3.1 Die Misch-Masch-Maschine

Die Mathematiker Dr. Heike Schuster und Dr. Martin Gerhardt entwickelten in den achtziger Jahren ein mathematisches Modell zur Simulation der räumlichen Strukturbildung einer oszillierenden Kohlenstoff-Oxidation. Bei diesem chemischen Prozess reagie-

¹Bosonen sind Elementarteilchen, die Kräfte übertragen

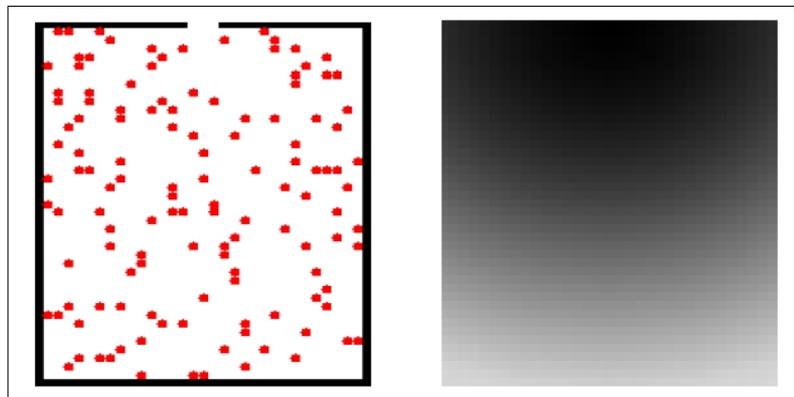


Abbildung 3.8: links: Gitter mit Fußgängern und Tür, rechts: Statisches Bodenfeld, das zur Tür hin immer stärker wird

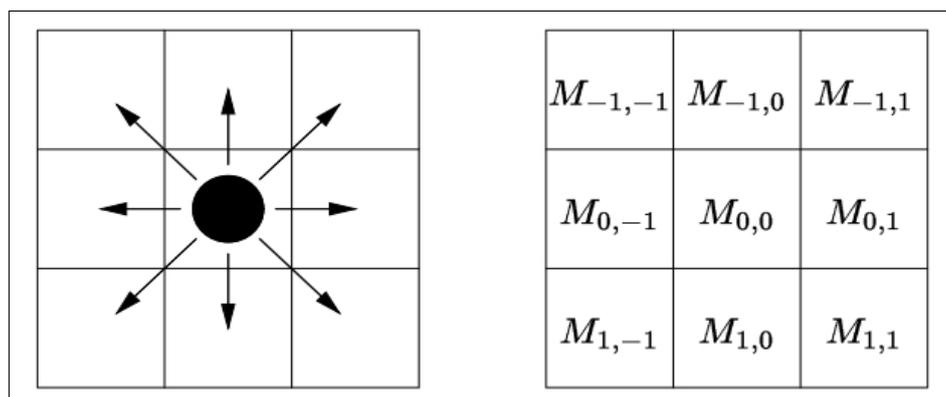


Abbildung 3.9: Präferenzmatrix

ren Kohlenstoff und Sauerstoff aufgrund eines Katalysators miteinander und produzieren dabei Kohlendioxid. Aus Versuchen war bekannt, dass die bei der Kohlenstoffoxidation produzierte Kohlendioxidmenge zeitlich schwankt. Einerseits konnten sie periodische Schwankungsmuster dokumentieren, aber auch komplexe, selbstähnliche Strukturen erkennen. Eine Annahme für diese Oszillationen war der Phasenumschlag zwischen katalytisch aktiven und inaktiven Phasen. Wissenschaftler aus den USA gelang dann, was vorher keiner geschafft hatte. Sie konnten Momente einer katalytischen Reaktion mittels einer Infrarotaufnahme sichtbar machen. Zu erkennen waren an der Katalysatoroberfläche räumliche Bereiche mit unterschiedlicher Temperatur. Diese Beobachtungen gaben den ersten Hinweis dafür, dass räumliche Musterbildungen die zeitlichen Strukturbildungen bedingen. [Gerhardt Schuster, 1995, S. 138ff]

Den Wissenschaftlern Schuster und Gerhardt schien es am besten, diese Aufgabe mit einem zellulären Automaten zu beschreiben. Das Ergebnis ihrer Arbeit nannten sie *Misch-Masch-Maschine*, da der Automat aus jedem beliebigen Chaos ein Muster erschafft. Sie interpretierten eine Zelle in einem zwei-dimensionalen Gitter als eine größere Menge von Palladium-Kristallen (Katalysator). Um den Phasenumschlag zwischen aktiven und inaktiven Phasen zu simulieren, wurde dafür ein Modell eines Infektionsprozesses angenommen. Eine aktive Phase wird durch eine gesunde Zelle dargestellt, eine inaktive durch eine kranke. Dazwischen befindet sich der Übergang, der mit dem Infektionsprozess gleichgesetzt wird.

Die Funktion F wird für verschiedene Zustände von $z_{ij}(t)$ unterschiedlich definiert [Gerhardt Schuster, 1995, S. 152] (Zusammenfassung s. Tab. 3.1):

- Eine *gesunde* Zelle kann durch kranke oder infizierte Nachbarn infiziert werden. Sei $K_{i,j}$ ($I_{i,j}$) die Anzahl der kranken (infizierten) Nachbarn der Zelle (i,j). Ist diese Zahl größer als ein Schwellwert k_1 (k_2), so wird diese Zelle infiziert.
- Eine *infizierte* Zelle passt sich im Grad ihrer Infektion durch eine lokale Mittelwertbildung ihren infizierten Nachbarn an und schreitet außerdem in jedem Zeitschritt um einen konstanten Betrag g ($g > 0$) voran. Ist $z_{ij}(t+1) > V$, wird der neue Zustandwert der Zelle (i,j) auf den Wert V gesetzt.
- Eine *kranke* Zelle wird sofort im nächsten Zeitschritt wieder gesund.

Zellraum	2-dimensional, m x n Gitter
Nachbarschaft	Moore, mit erweitertem Radius r
Zustandsmenge	{0,1,2,...,V} 0=gesunde Zelle, V=krankte Zelle, übrigen Werte stellen infizierte Zustände dar
Randbedingungen	beliebig
Anfangsbedingung	beliebig
Zustandsentwicklung	$z_{ij}(t + 1) = F(z_{ij}(t), N_{ij}(t)), \quad (3.8)$ <p>für $z_{ij}(t)=0$</p> $z_{ij}(t + 1) = \lfloor \frac{K_{ij}(t)}{k_1} \rfloor + \lfloor \frac{I_{ij}(t)}{k_2} \rfloor \quad (3.9)$ <p>für $0 < z_{ij}(t) < V$</p> $z_{ij}(t + 1) = \left[\left(\sum_{\substack{(k,l) \in N_{ij} \\ 0 < z_{kl}(t) < V}} z_{kl}(t) \right) / I_{ij}(t) \right] + g \quad (3.10)$ <p>für $z_{ij}(t)=V$</p> $z_{ij}(t + 1) = 0 \quad (3.11)$

Tabelle 3.1: Zusammenfassung Misch-Masch-Maschine [Gerhardt Schuster, 1995, S. 152]

Typische Muster der Misch-Masch-Maschine sind Kreis- und Spiralwellen (s. Abb. 3.10). Diese Ergebnisse wurden 1988 in einer Kolumne im Scientific American von Alexander Dewdney vorgestellt. Dieser wies auf die Ähnlichkeiten mit Mustern in der Belousov-Zhabotinsky-Reaktion hin. Keinem anderem theoretischen Modell war es bis zu diesem Zeitpunkt möglich, diese Ähnlichkeiten aufzuzeigen.

Anfang der neunziger Jahre gelang es dann Wissenschaftlern vom Fritz-Haber-Institut wären einer Kohlenstoff-Oxidation Spiral- und Kreiswellen zu beobachten. Im Jahre 2001 gelang es ihnen dann sogar die Musterbildung während einer CO-Oxidation zu steuern und zu beeinflussen. Somit war der Nachweis erbracht, dass es möglich war, chemische Prozesse durch Computereingriffe zu beeinflussen. [Stegemann, 2001]

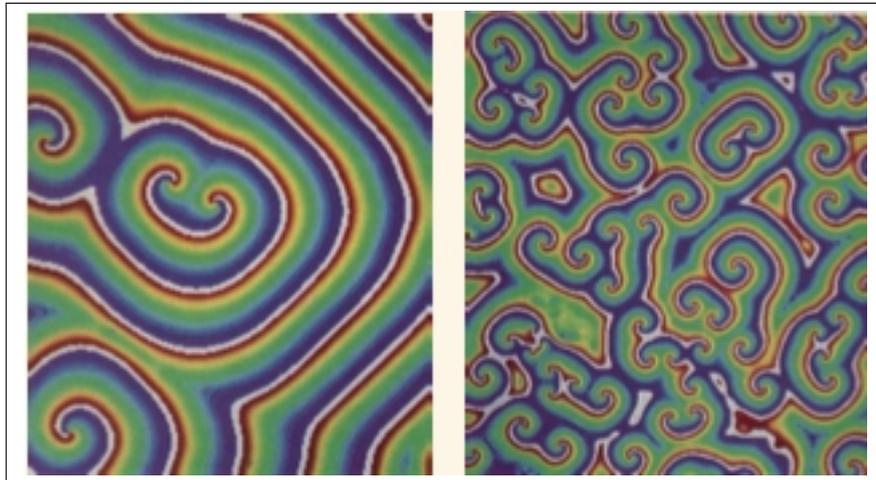


Abbildung 3.10: Muster der Misch-Masch-Maschine

3.4 ZA in der Physik

3.4.1 Zelluläre Gittergase

Bei einem Gittergas-Modell werden die Moleküle eines Gases als diskrete Werte an den Gitterpunkten eines regelmäßigen Gitters ausgerichtet. Die Teilchen können sich entlang der Gitterachse bewegen solange der Weg frei ist, andernfalls *stoßen* sie zusammen und ändern ihre Bewegungsrichtung laut einer zuvor definierten Entwicklungsregel. Dabei

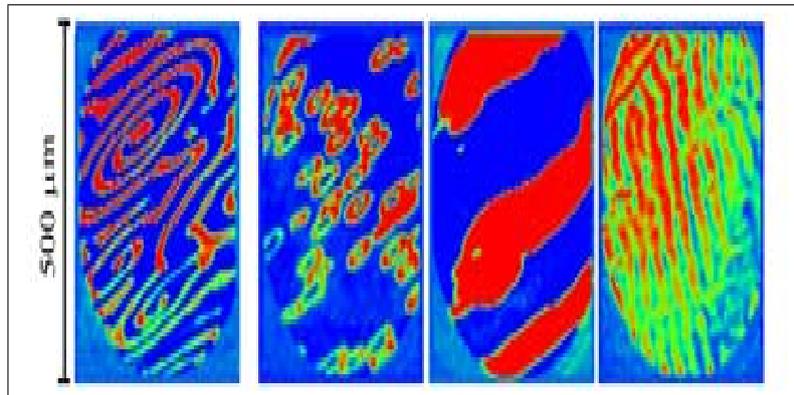


Abbildung 3.11: Muster während einer CO-Oxidation

gelten die grundlegenden Gesetze der Mechanik, wie die Erhaltung der Energie ² und des Impulses ³. Das bedeutet einerseits, dass die Anzahl der Teilchen in einem System immer konstant bleibt, sowie auch die Geschwindigkeit, mit der sich die Teilchen fortbewegen. Ein weiterer Aspekt, der bei diesem Modell berücksichtigt werden muss, ist die Reversibilität, das bedeutet die Umkehrbarkeit der Prozesse in der Zeit.

Für die Modellierung eines Gittergases mit Hilfe eines zellularen Automaten bedeuten die zuvor genannten Punkte folgendes:

- *Reversibilität*: Jeder Zustand einer Zelle hat einen eindeutig definierten Vorgänger.
- *Energie- und Impulserhaltung*: Wenn eine aktive Zelle ein Teilchen symbolisiert und eine inaktive Zelle einen Freiraum, dann bleibt die Anzahl der aktiven und inaktiven Zellen konstant. Die Geschwindigkeit mit der sich die Partikel fortbewegen bleibt ebenfalls konstant.
- *Bewegung*: Die Bewegung der Teilchen muss durch Entwicklungsregeln definiert werden.

Um die Verfolgung eines Teilchens nachzuvollziehen können, müssen sämtliche Zellzustände einer Nachbarschaft aktualisiert werden. Der Wissenschaftler N. Margolus hat hierfür die *Margolus-Nachbarschaft* definiert: Der Zellraum wird dazu partitioniert und in Blöcke mit einer Größe von 2×2 Zellen zerlegt. Nach jedem Zeitschritt verschiebt sich

²Energieerhaltungssatz: Die Gesamtenergie in einem System bleibt konstant. Energie kann nur umgewandelt werden, jedoch nicht erzeugt oder vernichtet werden.

³Impulserhaltungssatz: Der Gesamtimpuls eines Systems bleibt konstant.

dieses Raster diagonal um eine Zelle (s Abb. 3.12, 2x2 Blöcke sind mit der dicken Rahmenlinie gekennzeichnet) [Goos, 1998]. In Abb. 3.13 sind die 2x2 Blöcke für die Bewegung eines Teilchens entlang einer Diagonalen für die Zeitpunkte t und $t+1$ dargestellt. Die Randbedingungen eines Zellraums sind periodisch.

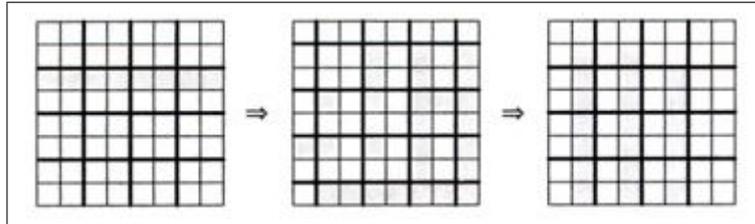


Abbildung 3.12: Blöcke der Margolus-Nachbarschaft

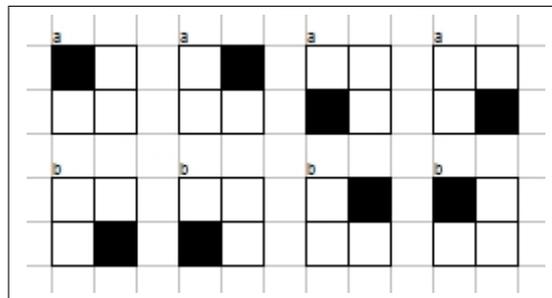


Abbildung 3.13: Bewegungsrichtung durch Blockregel a: $z(t)$, b: $z(t+1)$

Die Kollision zweier Teilchen ist in Abb. 3.14 dargestellt. Dabei handelt es sich um ein Beispiel aus dem Modell des *HPP-Gases*, welches nach dessen Erfinder J. Hardy, O. de Pazis und Y. Pomeau benannt wurde. Nach der Kollision ändern die Teilchen ihre Bewegungsrichtung um 90° . Weiter Entwicklungsregeln sind in Abb. 3.15 dargestellt. Durch die horizontale und vertikale Spiegelung an der Blockachse lassen sich somit alle möglichen Muster eines Blockes darstellen. Mit dem HPP-Modell ist es möglich, Strömungsverhalten von Gasen und Flüssigkeiten zu simulieren, wie z. B. die Ausbreitung einer Schallwelle. [Gerhardt Schuster, 1995, S. 116]

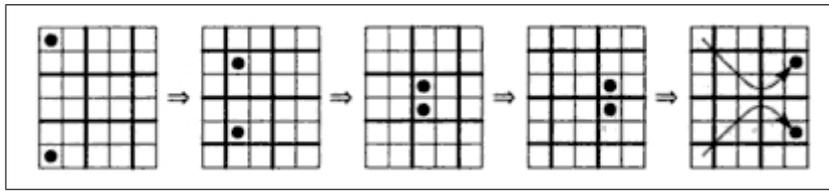


Abbildung 3.14: Zusammenstoß von zwei Teilchen im HPP-Gas

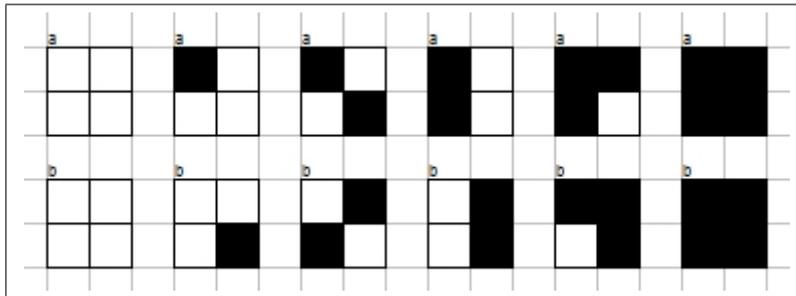


Abbildung 3.15: Entwicklungsregeln im HPP-Gas

3.5 ZA in der Biologie

3.5.1 Musterbildung: Aktivator-Inhibitor-Modell

Die Wissenschaftler Hans Meinhardt und Alfred Gierer entwickelten gemeinsam ein Modell zur natürlichen Musterbildung auf Basis von partiellen Differentialgleichungen. Das zugrundeliegende Prinzip beruht auf zwei gegenläufigen Kräften. Der *Aktivator* verstärkt den Aufbau von bestimmten Strukturen und der *Inhibitor* versucht genau das zu verhindern. In der Biologie werden diese Substanzen Morphogene⁴ genannt. Diese sind Signalmoleküle, die im Gewebe in unterschiedlichen Konzentrationen vorliegen. Sie steuern die Entwicklung der Musterbildung bei Lebewesen.

David Young entwickelte auf diesem Prinzip einen 2-dimensionalen zellulären Automaten, der als Ergebnis die in der Natur typischen Flecken- und Streifenmuster hat. Im einfachsten Fall geht man von einer zufälligen Verteilung von schwarzen und weißen Zellen in einem 2-dimensionalen Gitter aus, dabei entspricht eine Zelle im Automat einer Pigmentzelle. Aus biologischer Sicht entstehen die weißen oder schwarzen Zellen aufgrund der *Zelldifferenzierung*. Darunter versteht man einen langsamen, zufallsgesteu-

⁴Schon Alan Turing hat in den 50er Jahren das Prinzip der Morphogene behandelt (s. [Turing, 1952]), jedoch war es ihm zu damaliger Zeit nicht möglich, seine Theorie zu belegen. Aufgrund der Tatsache, dass er zum großen Teil recht hatte, werden diese Muster *Turingmuster* genannt

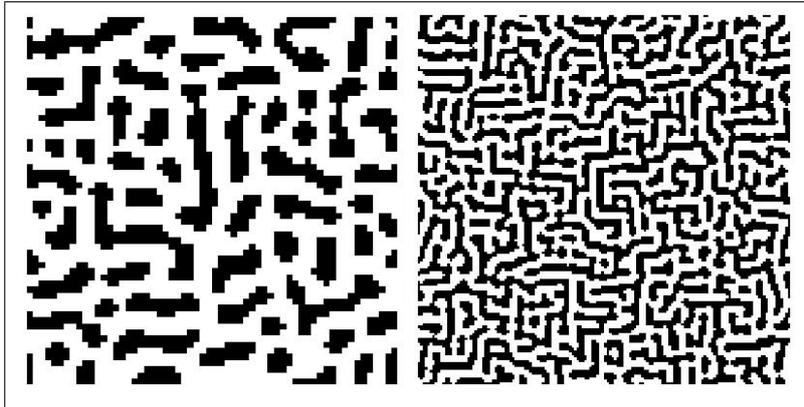


Abbildung 3.16: Typische Muster im AIM

erten Prozess, bei dem die undifferenzierten weißen Zellen zu differenzierten schwarzen Zellen werden. Sobald eine Zelle differenziert ist, kann diese zwei Kräfte bilden, nämlich den zuvor genannten Aktivator und Inhibitor, welche ihre Kräfte durch Diffusion in die Nachbarschaft verteilen können. Zum Unterschied zu bereits bekannten Nachbarschaften führt Young für sein Modell ein neues Prinzip ein. Er definiert den Nachbarschaftsraum einer Zelle als zwei kreisförmige Gebiete. Innerhalb des kleinen Kreises wirken Aktivator- und Inhibitorsubstanzen, jedoch ist die Kraft des Aktivators gegenüber des Inhibitors dominanter. In einem größeren Kreis ist aber nur mehr die hemmende Wirkung des Inhibitors wirksam, dessen Diffusionsgeschwindigkeit auch höher gegenüber des Aktivators ist. Zusätzlich werden im kleinen Kreis alle Zustände mit einem positiven Gewichtungsfaktor aufsummiert, hingegen im größeren Kreis mit einem negativen Gewichtungsfaktor. Die Zustandsentwicklung erfolgt durch das Zusammenzählen der unterschiedlichen Werte der Kräfte. Ist das Ergebnis positiv, wird bzw. bleibt die Zelle schwarz, andernfalls wird sie weiß.

Typische Muster, die sich durch die Zustandsentwicklung ergeben, sind in Abb. 3.16 dargestellt. Abhängig von der Stärke des Inhibitors, entstehen entweder Flecken oder Streifen. Es ist auch möglich, die Nachbarschaft als Ellipse zu definieren, dann entstehen langgezogene Streifenmuster.

Zellraum	2-dimensional, m x n Gitter
Nachbarschaft	Kreisförmiges Gebiet vom Radius R $N_{ij} = \{(k, l) (k - i)^2 + (l - j)^2 < R^2\} \quad (3.12)$
Zustandsmenge	{0,1}
Randbedingungen	beliebig
Anfangsbedingung	beliebig
Zustandsentwicklung	$z_{ij}(t + 1) = H\left(\sum_{(k,l) \in N_{ij}} w(k, l) * z_{kl}(t)\right) \quad (3.13)$ <p>wobei H für die Heavyside-Funktion steht und w einen Gewichtsfaktor darstellt:</p> $H(x) = \begin{cases} 1 & \text{wenn } x < 0 \\ 0 & \text{wenn } x \geq 0 \end{cases}$ $w(k, l) = \begin{cases} 1 & \text{wenn } (k - i)^2 + (l - j)^2 \leq R_a^2 \\ w_I & \text{wenn } R_a^2 < (k - i)^2 + (l - j)^2 \leq R^2 \end{cases}$

Tabelle 3.2: Zusammenfassung Aktivator-Inhibitor-Modell [Gerhardt Schuster, 1995, S. 208]

3.5.2 Der Hyperzyklus

Die Theorie des Hyperzyklus, postuliert vom Biochemiker Manfred Eigen, beschreibt Wachstums- und Rückkopplungsfunktionen im vorbiologischen Evolutionsprozess⁵. Der Begriff steht für die *molekulare Selbstorganisation* von Nukleotidketten und Proteinen, den Grundbausteinen von Zellen. Jeder einzelne Zyklus besteht aus Polynukleotiden⁶ und Proteinmolekülen⁷. Mehrere Zyklen bilden wiederum einen Hyperzyklus (s. Abb. 3.17). Bei den zugrundeliegenden Experimenten konnte beobachtet werden, dass sich spontan autokatalytische Prozesse ausbilden. Bei längerer Entwicklung schließen sich mehrere autokatalytischen Strukturen zu einem Hyperzyklus zusammen. Der große Vorteil liegt in der Korrektur von Replikationsfehlern. Fehlerhafte Hyperzyklen mutieren entweder vorteilhaft, oder sie fallen auseinander. Die freiwerdenden Moleküle dienen als Grundbausteine für neue Hyperzyklen.

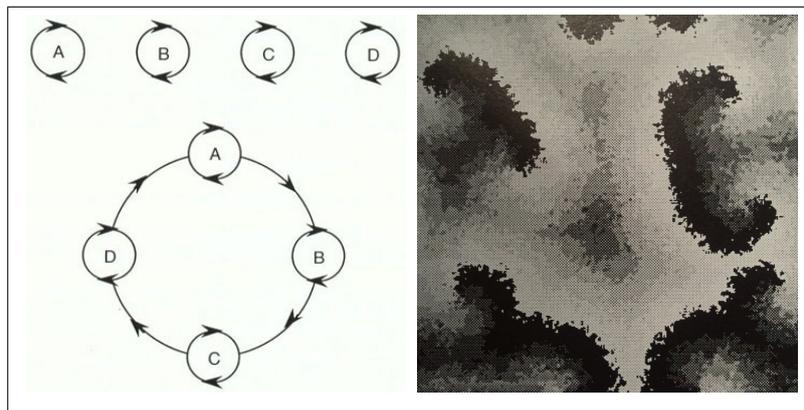


Abbildung 3.17: Entwicklungsschleife und Musterbildung des Hyperzyklus

Das zelluläre Modell des Hyperzyklus

Als Grundlage dient eine 2-dimensionale Gitter mit periodischen Randbedingungen. Anfangs sind 50% der Zellen mit unterschiedlichen Molekülarten besetzt. Jede Molekülart

⁵Die vorbiologische Phase der Evolution beschreibt die Entwicklung von einer Vorstufe des Lebens auf molekularer Ebene.

⁶Polynukleotide sind mehrere miteinander verbundene Nukleotide. Diese sind aus einem Phosphat-, einem Zucker- und einem Basenbestandteil zusammengesetzt. Polynukleotide dienen als Informationsträger in Genen als Bestandteil der DNA.

⁷Proteine sind Moleküle aus Aminosäuren. Sie befinden sich in Zellen und dienen unter anderem als Katalysatoren für chemische Reaktionen.

Zellraum	2-dimensional, m x n Gitter
Nachbarschaft	Moore-Nachbarschaft
Zustandsmenge	$\{0,1,\dots,M\}$
Randbedingungen	periodisch
Anfangsbedingung	beliebig
Zustandsentwicklung	$z_{ij}(t+1) = DIFF^2(H(z_{ij}(t), N_{ij}(t))) \quad (3.14)$ wobei DIFF den Algorithmus von Toffoli und Margolus zur Diffusion beschreibt ¹ .

¹ Wie beim HPP-Gas (s. Kap. 3.4.1) wird das Gitter in 2 x 2-Blöcke unterteilt. Die Zellen werden innerhalb eines Blockes bewegt. Der Block wird zu jedem Zeitschritt t+1 mit der gleichen Wahrscheinlichkeit entweder in oder gegen den Uhrzeigersinn gedreht.

Tabelle 3.3: Zusammenfassung Der zelluläre Automat im Hyperzyklus [Gerhardt Schuster, 1995, S. 182]

wächst in Richtung ihres katalytischen Unterstützers (s. Abb. 3.17): Zellen B suchen Zellen mit dem Wert A, Zellen D orientieren sich in Richtung Zellen mit dem Zustand C. Sind mindesten fünf unterschiedliche Molekülarten vorhanden, ergeben sich räumliche Spiralmuster, die innerhalb von ungefähr 2000 Zeitschritten Ordnung in den chaotischen Anfangszustand bringen [Gerhardt Schuster, 1995, S. 187].

Die Regeln des zellulären Automaten im Hyperzyklus [Gerhardt Schuster, 1995, S. 185]:

- *Zerfall*: Jede besetzte Zelle zerfällt mit einer bestimmten Wahrscheinlichkeit zum Zeitpunkt t+1.
- *Selbstreplikation*: Mit einer bestimmten Wahrscheinlichkeit wird zum Zeitpunkt t+1 eine Zelle besetzt, wenn einer ihrer Nachbarn besetzt ist.
- *Katalytische Replikation*: Die Wahrscheinlichkeit, dass eine leere Zelle zum Zeitpunkt t+1 besetzt wird, erhöht sich, wenn der Vorgänger im Hyperzyklus einer ihrer besetzten Nachbarn zum Zeitpunkt t in Richtung der Replikation zu finden ist.

3.5.3 Der Planet Wator - Biologisches Modell eines Ökosystems

Die künstliche Welt des *Planeten Wator*⁸ wurde von Alexander Dewdney und David Wiseman entwickelt. Der Planet besteht zur Gänze aus Wasser, die einzigen Lebewesen sind kleine Fische und Raubfische, wie Haie. Das Grundprinzip lautet: fressen oder gefressen werden. Dem Modell des ZA liegt ein 2-dimensionales Gitter mit periodischen Randbedingung zugrunde. Es wird die von Neumann-Nachbarschaft angenommen und insgesamt gibt es drei unterschiedliche Zustände. Entweder ist eine Zelle leer, oder besetzt mit einem kleinen Fisch oder einem Hai. Für beide Fischarten gelten unterschiedliche Regeln. Ein kleiner Fisch kann sich in eine der vier Hauptrichtungen bewegen, falls mindestens eine der vier Nachbarzellen leer ist. Allein der Zufall entscheidet über die Bewegungsrichtung. Sind alle Zellen besetzt, bleibt der Fisch zum Zeitpunkt $t+1$ in seiner Zelle. Existiert der Fisch lange genug auf dem Planeten (Parameter wird vorher festgelegt), kann sich der Fisch reproduzieren. Er bewegt sich einfach in eine Nachbarzelle und hinterlässt einen Nachkommen in seiner alten Zelle. Was die Reproduktion der Haie angeht, gelten die selben Regeln. Haie müssen jedoch, um zu überleben, nach Fischen suchen, die sie fressen können. Finden Haie eine bestimmte Zeit lang kein Futter, sterben sie. Der Hai sieht sich in seiner Umgebung um und falls er einen kleinen Fisch findet, frisst er diesen. Es wird einfach die Zelle des kleinen Fisches mit einem Hai besetzt. Befindet sich kein kleiner Fisch in unmittelbarer Nähe des Haies, bewegt sich dieser genauso zufällig fort wie der kleine Fisch.

Parameter, die von der Benutzerin festgelegt werden (ein Zyklus entspricht einem diskreten Zeitschritt, ein Existenzzyklus gibt die Anzahl der Zustände an, die eine Zelle aktiv war):

- *nfish*: Anzahl der kleinen Fische zu Beginn.
- *nshark*: Anzahl der Haie zu Beginn.
- *fbreed*: Anzahl der Existenzzyklen eines kleinen Fisches, bevor sich dieser Reproduzieren kann.
- *sbreed*: Anzahl der Existenzzyklen eines Haies, bevor sich dieser Reproduzieren kann.

⁸So auch die Entstehung des Namens Wator, einerseits das Wasser, andererseits die TORoidale Welt.

- *starve*: Anzahl der Zyklen indem der Hai Futter finden muss, sodass er nicht stirbt.

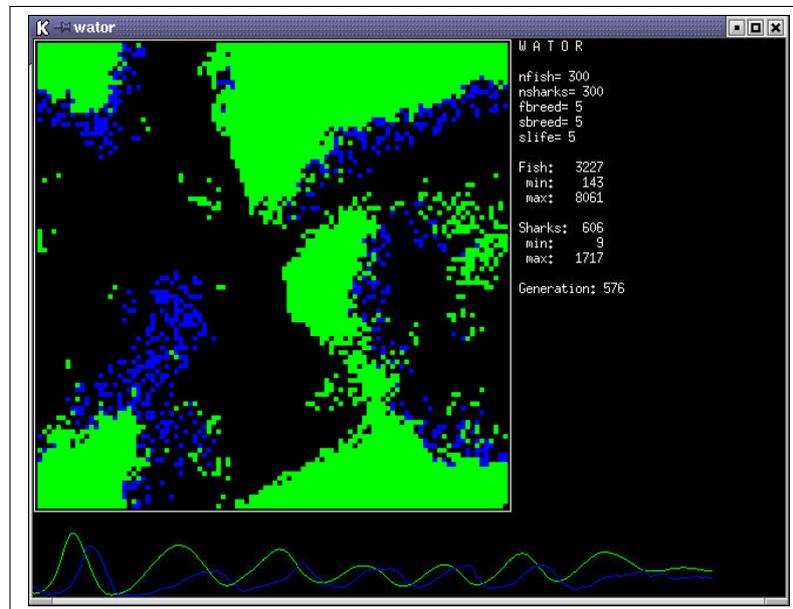


Abbildung 3.18: Simulation Planet Water

Diese einfachen Regeln bieten eine Vielzahl von Spielmöglichkeiten und besitzen auch Entwicklungspotential. Als Beispiel kann hier genannt werden, dass Fische theoretisch ebenfalls fressen müssten, um zu überleben, jedoch wird in diesem einfachen Modell das Vorhandensein von Plankton als Gegeben angenommen. Abhängig von den voreingestellten Parametern, entsteht ein ökologisches Gleichgewicht oder das große Aussterben beginnt. Im besten Fall stellen sich unterschiedliche Populationszyklen ein, einmal mit mehr und weniger Mitgliedern ihrer Art. Zu erkennen ist auch eine gewisse Logik, dass Haie zum Beispiel nie alle Fische auffressen dürfen, sonst fehlt ihnen die Lebensgrundlage und auch sie sterben aus.

Zellraum	2-dimensional, m x n Gitter
Nachbarschaft	von-Neumann-Nachbarschaft
Zustandsmenge	$\{(f, h, v) 0 \leq f \leq R_f, 0 \leq h \leq R_h, 0 \leq v \leq V\} \quad (3.15)$ <p>f: Alter der Fische h: Alter der Haie v: mögliche Hungerperiode der Haie R_f, R_h: Reproduktionsperiode der Fische bzw. Haie V: max. Hungerzeit der Haie</p>
Randbedingungen	periodisch
Anfangsbedingung	beliebig
Zustandsentwicklung	<ol style="list-style-type: none"> 1. Suche nach freien Plätzen 2. Bewegung und Fortpflanzung $f_{ij}(t+1) = \begin{cases} 1 & \text{wenn } f_{ij}(t) = R_f \\ 0 & \text{wenn } sonst \end{cases}$ $f_{kl}(t+1) = \begin{cases} 1 & \text{wenn } f_{ij}(t) = R_f \\ 0 & \text{wenn } sonst \end{cases}$ <ol style="list-style-type: none"> 3. Alterung $f_{ij}(t+1) = \begin{cases} f_{ij}(t+1) & \text{wenn } 0 < f_{ij}(t) < R_f \\ R_f & \text{wenn } f_{ij}(t) = R_f \\ 0 & \text{wenn } f_{ij}(t) = 0 \end{cases}$

Tabelle 3.4: Zusammenfassung WATOR [Gerhardt Schuster, 1995, S. 220f]

Kapitel 4

Zelluäre Automaten in der Musik

Im folgenden Kapitel soll ein Überblick über Implementationen und Versuche über den Einsatz ZA im Bereich der musikalischen Strukturgenese gegeben werden. Künstler auf diesem Gebiet sind nicht nur Komponisten im klassisch musikalischen Sinn, sondern besitzen auch einen Background im Bereich der Informationstechnik (wie z.B.A. Dorin, P. Beyls, u.a.). Die Synthese zwischen diesen beiden Disziplinen zeigt meiner Meinung nach eine bemerkenswerte Herangehensweise an das künstlerische Schaffen.

Um mit ZA im künstlerischen Sinn arbeiten zu können, muss im Vorfeld die zugrundeliegende Theorie mit dem gewünschten musikalischen Output in Verbindung gebracht werden. Aus meiner Sicht wird schnell klar, dass mit einer Steigerung der Komplexität des Automaten auch die Nachvollziehbarkeit der daraus resultierenden musikalischen Struktur erschwert wird. Das Hauptaugenmerk liegt am Ende im Mapping der Zellen, der Zustände oder des Automaten im Gesamten. Wie wird es möglich, das simulierte Geschehen in Musik umzuwandeln und dabei den Bezug zur Regel, zur Zustandsänderung, zum Automaten selbst zu erhalten und hörbar zu machen? Die sinnvolle, einer bestimmten Ästhetik entsprechenden Hörbarmachung der Daten selbst ist der Ausschlaggebende Punkt bei der Generierung musikalischer Struktur durch ZA.

4.1 Eleonora Bilotta - Musical Dreams

Eleonora Bilotta, Pietro Pantano und Valerio Talarico untersuchten in [Bilotta Pantano Talarico, 2000] die Eignung 1-dimensionaler Automaten zur Generierung von musikalischen Mustern. Insbesondere wurden die Automaten der einzelnen vier Klassen Wolframs untersucht.

Mit der Software *Musical Dreams*, die zur Simulation und zum Rendering dieser Automaten dient, wurden die entstandenen Muster analysiert.

Um die Komplexität der ZA in Musik übertragen zu können, entwarfen Billota et al. einen semiotischen Analyseansatz. Daraus entstanden folgende Anforderungen [Bilotta Pantano, 2001]:

- *A code establishes the correlation between a plan of the expression (in its formal and systematic aspects) and a plan of the content (in its formal and systematic aspects).*
- *A semiotic function establishes the correlation between an element taken from the expression system and an element taken from the contents system.*
- *In this way a code establishes general types thus producing the rule that generates the concrete occurrence (in this case musical signs).*
- *Normally both continuums represent the elements that proceed semiotic correlation, but in the case of cellular automata, expressing themselves graphically they maybe considered intrinsically as semiotic material. The same can be said for the digital continuum that realizes the music physically.*

Aus diesen Anforderungen entstanden wiederum Ideen, die unabdingbar wurden für eine vernünftige Kodierung zellulärer Daten. Das System muss so konfiguriert werden, dass die daraus entstandene Komposition auf den Automaten zurückgeführt werden kann.

Mit Hilfe dieser Überlegungen entwickelten sie unterschiedliche Codes und Mapping-Techniken zur Generierung musikalischer Strukturen mit Hilfe eines ZA. Unterschieden werden drei Arten:

- *Local Musification Codes* (Bog, Colour Code, Minuet), through which it is possible to read the grid of a CA site by site.
- *Global Musification Codes* (Input-Entropy, Populations), through which the functions of input entropy and the progress of the population by CAs.
- *Mixed Musification Codes*, reading portions of CA configurations either with binary codes or with color codes.

Im folgenden sind einige Local Musification Codes (LMC) kurz zusammengefasst:

- *BOG*: In diesem Fall stellen die horizontalen Zellen eine Klaviertastatur dar (s. Abb. 4.1). Wenn eine Zelle aktiv ist, wird die Taste gespielt, ist eine Zelle nicht aktiv, wird in diesem Fall pausiert. Besitzt der Automat mehr als nur zwei Zustände, kann der Wert der Zelle als Dauer der Note interpretiert werden.

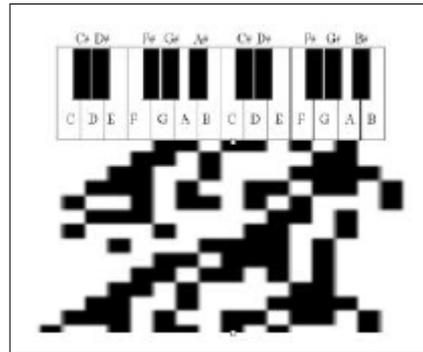


Abbildung 4.1: Mapping in BOG

- *COLOUR CODE*: In diesem Fall werden die unterschiedlichen Zustände der Zellen durch Farben repräsentiert. Anschließend wird jeder Farbe ein Ton zugeordnet. Der einzige Nachteil dieser Mapping-Methode ist die immer gleiche Dauer der Töne. Dies könnte aber noch durch einen zusätzlichen Parameter gesteuert werden.
- *MINUET*: Hier wird das Gitter in Blöcke von 3 x 3 Zellen unterteilt. Jede Zelle besitzt einen Zustand k . Die Werte der einzelnen Zellen eines Blocks werden aufaddiert, und der resultierende Wert eines Blocks einem musikalischen Parameter zugeordnet. So bestimmt z.B. der erste Block die Tonhöhe, der zweite Block die Tondauer usw..

Der Klang der Glider mit den LMC

Wie schon bei Conways *Game of Life*, wurden sogenannte *glider* entdeckt und näher untersucht. Dies können Organismen sein, welche sich mit einer bestimmten Periode wiederholen und vertikal oder diagonal über den Bildschirm wandern. Für die Regel 20 entstehen je nach unterschiedlicher Anfangskonfiguration (IC = Initial Condition) auch unterschiedliche *glider*.

Gliders, die diagonal über den Bildschirm wandern, besitzen ein stetes Tempo. Die Bewegung des Organismus nach rechts oder links bewirkt eine Transposition der Phrase zu einem höheren oder niedrigeren Halbton. *Gliders* wiederum, die in ihrer Position verharren, klingen monoton, da keine Veränderung auftritt.

Abschließend kann man sagen, dass Bilotta und ihre Kollegen zu folgenden Erkenntnissen bezüglich der Eignung 1-dimensionaler Automaten zur Generierung von musikalischen Mustern kamen:

- ZA, die zur ersten Klasse gehören, eignen sich gut um rhythmische Muster zu generieren
- ZA, die zur zweiten Klasse gehören, produzieren bessere harmonische Muster
- Chaotische Systeme ergeben nur brauchbare Ergebnisse, wenn sie mit simplen Anfangskonfigurationen gestartet werden.

Anmerkungen zu Bog, Colour Code und Minuet

Bei dem Mappingalgorithmus *Bog* wird die horizontale Zellebene einer Klaviertastatur zugeordnet. Dies stellt zwar eine sehr klassische Möglichkeit dar, Klangfolgen zu generieren, jedoch suggeriert dieser Algorithmus, dass der Automat inhaltlich aufgrund der steigenden Tonhöhe von links nach rechts zu interpretieren ist. Da aber jede Zelle eines ZA gleichwertig ist und nur dessen Entwicklung in der Zeit den Inhalt des Automaten vermittelt, suggeriert diese Mappingstrategie ein falsches Bild.

Der *Colour Code* ordnet jeden Zustand des Automaten einer bestimmten Tonhöhe zu. Je mehr Zustände eine Zelle annehmen kann, desto komplexer ist die Entwicklung des Automaten. Da in diesem Fall nur die Zustände gemappt werden, geht einerseits die zeitliche Entwicklung der einzelnen Zellen verloren als auch die räumliche Struktur des Automaten.

Bei dem Algorithmus *Minuet* wird der 2-dimensionale Zellraum in Blöcke von 3 x 3 Zellen unterteilt. Die Werte der einzelnen Zellen werden aufaddiert und musikalischen Parametern zugeordnet. In diesem Fall geht wieder die Information der zeitliche Entwicklung der einzelnen Zellen verloren. Problematisch ist auch die grundsätzliche Teilung

des Gitters in festgelegte Blockgrößen, da hier wiederum der Bezug zur jeweiligen Regel fehlt.

4.2 Alan Dorin - Liquiprism

Liquiprism ist ein auf MIDI basierendes Instrument, welches polyrhythmische Strukturen generiert. Es kann einerseits als autonomes Instrument in einer Installation eingesetzt werden, oder auch als Instrumente zur generativen Komposition.

The work is intended to focus on the seeming conflict between the determinism and regularity or cellular automata, and the apparently complex and fluid changes of form and pattern their interactivities produce. [Dorin, 2002]

Liquiprism besteht aus einem Würfel, dessen sechs Oberflächen jeweils einen Zellulären Automaten beinhalten. Die Moore-Nachbarschaft der Zellen wird an den Kanten des Würfels erweitert auf jeweils die nächste Oberfläche, somit ergibt sich eine geschlossene toroide Oberfläche. Das Update einer Zelle in einen neuen Zustand ist zeitlich auf jeder Oberfläche unterschiedlich geregelt. Somit ergibt sich für eine Zelle, die über eine Kante wandert, eine variierende update-Geschwindigkeit. Die Anzahl der unterschiedlichen Zustände, die eine Zelle annehmen kann, wird auf ON und OFF beschränkt.

Es gibt zwei unterschiedliche Sets an Transitionsregeln, die zu bestimmten Zeitpunkten zum Einsatz kommen. Das erste Set besteht aus folgenden zwei Regeln:

- Eine Zelle, die ON ist bleibt auch ON, wenn genau 2 oder 3 ihrer Nachbarzellen ebenfalls ON sind.
- Eine Zelle die OFF ist, bleibt OFF, außer mindestens 4 ihrer Nachbarzellen befinden sich im Zustand ON.

Das zweite Set besitzt die selbe Regel bezüglich Zellen die im Zustand ON sind, jedoch wird für die Zelle im OFF folgende Regel angewandt:

- Eine Zelle die OFF ist, wechselt im nächsten Zeitschritt in den Zustand ON, wenn sich die Zelle unterhalb im Zustand ON befindet. (Hat zur Folge, dass Zellen nach „oben“ wandern können).

Grundsätzlich wird das erste Set angewandt, pendelt sich jedoch ein periodischer Zustand ein oder es kommt zu einem Stillstand im System, wird das zweite Set angewandt um Muster zu generieren, die erneut über das Gitter wandern. Es besteht auch noch die Möglichkeit, eine Zustandsänderung der Zellen durch den Anwender per Mausklick herbeizuführen.

Das Mapping in Liquiprism

Das Ziel dieses Mappings ist die Generierung von komplexen rhythmischen Strukturen anhand einfacher Elemente. Um diesem Kriterium gerecht zu werden, werden die aktiven Zellen, welche zur Generierung musikalischer Strukturen dienen, auf einige wenige pro Zeiteinheit beschränkt. Es werden nur diejenigen Zellen verwendet, die vom Zustand OFF in den Zustand ON wechseln. Jede der sechs Oberflächen des Würfels ist mit einem MIDI – Kanal verbunden und schickt durch seine aktiven Zellen Steuerdaten an das angeschlossene Gerät. Um zu vermeiden, dass du viele Zellen Steuerdaten schicken, gibt es eine Begrenzung auf zwei Zellen pro Würfelseite. Jeder Zelle wird eine Tonhöhe zugeordnet, horizontal in Halbtonschritten folgend und vertikal im Quintabstand. Es besteht ebenfalls die Möglichkeit, andere MIDI Parameter, wie *Dauer* oder *Anschlagstärke* anzusteuern. Die Entscheidung liegt alleine beim Anwender.

Zusammenfassend kann gesagt werden, dass hier ein Versuch unternommen wurde, den Algorithmus hinter der Musik nicht nur direkt hörbar, sondern auch sichtbar zu machen. Den Entwicklern war es wichtig, dass ein logischer Zusammenhang auch für den Zuhörer zu erkennen ist, der nicht unbedingt den genauen Hintergrund eines ZA kennt.

Anmerkungen zu Liquiprism

Der Vorteil dieser Umsetzung ist, dass Zuhörer durch den Würfel eine Idee davon bekommen, wie Klänge generiert werden. Die einfachste Assoziation stellt in diesem Fall das Leuchten einer Zelle mit dem Erklängen eines Tones dar. Da die Anzahl der Zellen jedoch beschränkt ist, die zur Generierung von Tönen herangezogen werden, ist es schwierig zu erkennen, welche Zelle nun wirklich aktiv das Klanggeschehen beeinflusst. Ob es möglich ist, dass durch den Würfel wirklich klar ist, welche Regel dem Automaten zugrunde liegt, bleibt meiner Meinung nach nur erfahrenen Künstlern vorbehalten, die sich mit der Thematik der ZA schon ausführlich auseinandergesetzt haben.

4.3 Hunt, Kirk, Orton

Cellular Automata Workstation

CA can be conveniently used to produce large amounts of patterned data with a varying amount of self-similarity. This fact has led many people to suspect that they may be readily translatable into musical material because music can be regarded as a series of sounds related to each other in various ways and thus exhibiting structural coherence.[Hunt Kirk Orton, 1991]

Das Programm *Cellular Automata Workstation (CAW)* wurden von den Künstlern Ross Kirk und Richard Orton zusammen mit dem Software-Programmierer Andy Hunt entwickelt. Im Vordergrund stand dabei, dass die Generierung von Musik durch ZA kein automatisierter Prozess ist, sondern der/die KünstlerIn aktiv in das System eingreifen kann und somit das künstlerische Geschehen auch steuern kann.

Aufbau der Software

Am Startbildschirm können Gittergröße, Nachbarschaftsradien, Anzahl der Zustände und Regeln ausgewählt werden. Die grafische Ausgabe des zellularen Automaten erfolgt sobald das Programm mit den gewünschten Parametern gestartet wurde. Werden interessante Muster entdeckt, können Teile des Musters vergrößert und die Zellen musikalisch gemappt werden. Das so entstandene graphische Muster kann beliebig viele Zeitschritte durchlaufen, den Anfang und das Ende bestimmt allein der/die AnwenderIn.

Das musikalische Mapping

Der Anwenderin stehen mehrere Möglichkeiten zur Mapping der Daten zur Verfügung. Die einfachste Möglichkeit stellt das sogenannte *pitch-set* dar. Dabei wird jeder horizontalen Zelle am Bildschirm eine Tonhöhe zugewiesen. Wenn eine Zelle aktiv ist, erklingt ein Ton. Entweder werden die Zeitschritte automatisiert durchlaufen oder die Anwenderin selbst bestimmt das Tempo. Ebenfalls besteht die Möglichkeit in der Zeit zurückzugehen und Klänge erneut abzuspielen. Somit wird die *Workstation* zu einem *Live-Performance-Tool*. Zusätzlich können Lautstärke und Instrument beliebig während der Performance verändert werden. Durch das sogenannte *pitch-mask* können einzelne Zellen auch stumm

geschaltet werden, sodass eine Reduktion des Klanges erfolgt.

Eine weitere Mapping-Strategie erlaubt der Anwenderin, das entstandene Muster in Blöcke zu unterteilen und die Werte der Zellen als Steuerparameter für weitere Syntheseprogramme zu verwenden. Weiters ist die Kopplung zum Programm *MidiGrid*¹ möglich.

...we recognize that aesthetically pleasing visual patterns do not automatically lead to interesting sound. Indeed the CA workstation was produced in order that composers could explore this interrelationship. It is possible to regard the CA patterns as providing an iconic representation of material being developed.[Hunt Kirk Orton, 1991]

Anmerkungen zu CAW

Die Erkenntnis, dass interessante Muster nicht automatisch zu interessanten Klängen führen, ist meiner Meinung nach essentiell. Durch die Möglichkeit, in der Zeit zurückzugehen und Klänge erneut abzuspielen, auch mit unterschiedlichen Tonhöhenmappings, sind Möglichkeiten gegeben, um das gleiche Muster unterschiedlich hörbar zu machen. Dadurch wird auch ersichtlich, welche Mappings musikalisch Sinn machen und welche nicht.

4.4 Dale Millen - Cellular Automata Music

Der Komponist Dale Millen entwickelte die erste Version von *Cellular Automata Music (CAM)* in den Jahren 1988 bis 1990 für die damals gängigen MAC Betriebssysteme. Um die Kompatibilität zu neueren Geräten zu gewährleisten, wurde 2004 eine neue Version entwickelt, die nun auch aktuelle MIDI-Geräte per USB-Anschluss unterstützt.

Diese interaktive Anwendung ermöglicht es der Künstlerin, aufgrund der grafischen Ausgabe des ZA künstlerisch in das Geschehen einzugreifen. Insgesamt gibt es drei Betriebszustände in CAM: den *Run Mode*, *Edit Mode* und den *Play Mode*. Bevor ein Ablauf gestartet wird, muss von der Anwenderin zuerst die Anzahl der Zustände und der Nachbarschaftsradius definiert werden. Unterschieden wird zwischen *K3R1*, *K2R1* und *K2R2*.

¹Midigrid besteht aus 200 Blöcken, die mit einer Computermaus oder einem Joystick angeklickt werden können. Jedem Block wird ein Ton oder ein Akkord zugeordnet und kann dar per Mausclick abgespielt werden. [www.midigrid.fullpitcher.co.uk/index.html]

Grundsätzlich entspricht ein Pixel am Bildschirm einer Zelle, jedoch können die Pixellanzahl pro Zelle auf 2, 5, 10 oder 20 angepasst werden.

Run Mode, Edit Mode, Play Mode

Im *Run Mode* wird der zellulare Automat mit den zuvor definierten Parametern grafisch ausgegeben. In diesem Betriebszustand wird der Künstlerin ermöglicht, direkt in das Geschehen einzugreifen, indem sie aktiv die Parameter während der grafischen Ausgabe des Automaten nach belieben anpassen kann. Werden Einstellungen gefunden, die interessante musikalische Strukturen ergeben, können entweder die Parameter oder auch die entstandene Musik abgespeichert werden. Im *Edit Mode* kann die Entwicklung des zellularen Automaten Schritt für Schritt angepasst werden. Nach jedem Zeitschritt können Parameter verändert und auch abgespeichert werden. Im *Play Mode* können die zuvor abgespeicherten Parameter oder Musiksequenzen wieder aufgerufen und abgespielt werden. So wird es ermöglicht, musikalische Strukturen einzeln zu generieren und sie danach zusammensetzen.

Mapping

Jede Spalte kann einer der 16 ansteuerbaren MIDI-Kanäle zugeordnet werden. Die Kanäle können einzeln ein- und ausgeschaltet werden. Inaktive Zellen werden automatisch als Pausen ausgegeben.

Dave Millen war bei der Programmierung seiner Software wichtig, dass auf grafischen Muster eines zellularen Automaten in Form von musikalischem Output reagiert werden kann.

Certain cellular automata have vertical cellular formations containing patterns that repeat with specific periods. These formations are several cell columns in width. Each individual column may have a different short repeating pattern of state values that can be used by the composer to create a musical passage by combining the pattern in parallel with other patterns in the formation or by employing the pattern sequentially with other patterns in the formation [Millen, 2004]

Anmerkungen zu CAM

Der Vorteil dieser Software ist, dass während der Musikgenerierung direkt Parameter verändert und angepasst werden können. Auch hier wird ersichtlich, wie bei *CAM*, dass interessant aussehende Muster nicht direkt zu interessanten Klängen führen. Durch das stetige Eingreifen in das Geschehen, ist es aber dennoch möglich, musikalisch sinnvolle Strukturen zu generieren.

4.5 Eduardo Reck Miranda

Computers can compose if programmed properly. [Miranda, 1998]

Der Komponist und Informatiker E. R. Miranda wurde 1963 in Brasilien geboren. Nach seinem Informatik- und Kompositionsstudium begann er sich intensiv mit ZA als Kompositionsmittel und der Thematik der künstlichen Intelligenz auseinanderzusetzen. Miranda entwickelte zwei Computerprogramme - *Camus* und *Chaosynth* - die im folgenden Kapitel näher beschrieben werden. Beide dienen zur Generierung musikalischer Strukturen mit Hilfe von Computersoftware.

4.5.1 Camus

Das Computerprogramm *Camus* basiert auf zwei unterschiedliche Algorithmen. Einerseits auf den *Game of Life*-Algorithmus (s. Kap. 2.4) und andererseits auf das Programm *Demon Cyclic Space* (DCS) von David Griffeath. Letzteres basiert auf einem 2-dimensionalen Automaten mit k Zuständen. Die verschiedenen Zustände werden durch unterschiedliche Farben repräsentiert. Die Anfangskonfiguration besteht aus einem $n \times m$ Gitter mit zufällig angeordneten Zuständen. Jede Zelle zum Zeitpunkt t dominiert die Zelle zum Zeitpunkt $t-1$, das bedeutet, dass die benachbarten Zellen den Zustand t von der Zelle übernehmen. *Camus* verwendet ein kartesisches Koordinatensystem um einen Dreiklang darzustellen. Die x -Achse des Systems repräsentiert das erste Intervall und die y -Achse das Folgende.

Die zugrundeliegenden Algorithmen werden parallel verwendet. GoL-Organismen produzieren die Dreiklänge und die Muster im DCS bestimmen die Orchestrierung. Jede Farbe wird dabei einem bestimmten MIDI-Instrument zugewiesen. Die Dauer der einzelnen Klänge kann individuell bestimmt werden.

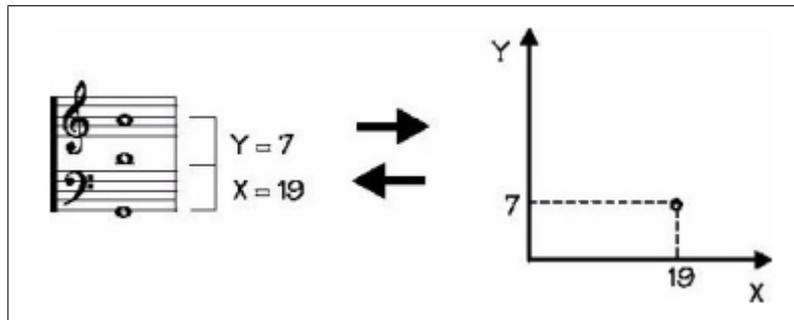


Abbildung 4.2: Mapping eines Akkordes in Camus

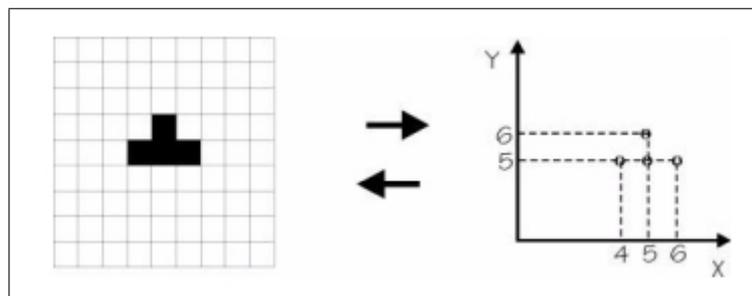


Abbildung 4.3: Mapping von aktiven Zellen in Camus

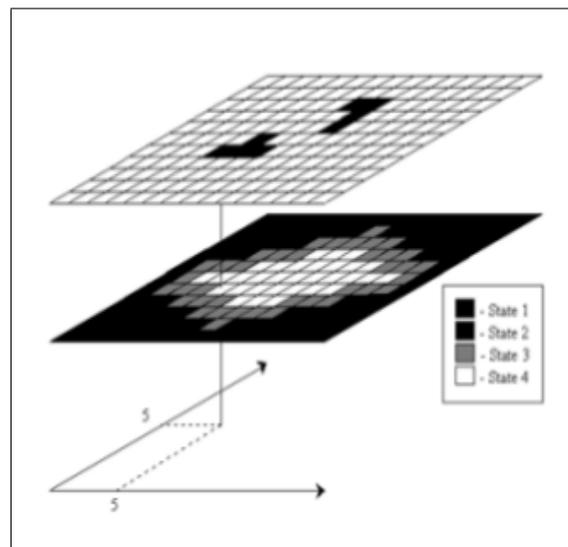


Abbildung 4.4: Klangerzeugung in Camus

Die Anfangskonfiguration von dem GoL-Gitter beginnt mit einer ausgewählten Startmuster, hingegen das DCS-Gitter mit einer zufälligen Anordnung der einzelnen Zellen und Zuständen. In Abb. 4.4 ist ein Beispiel gegeben das folgendermaßen interpretiert wird: die lebendigen Zellen der GoL-Ebene werden analysiert und die Dreiklänge ermittelt. Die korrespondierenden Zellen in der DCS-Ebene (besitzen die selben x/y -Koordinaten wie die lebendigen Zellen im GoL-Gitter) bestimmen dann die zugehörigen Instrumente. In Abb. 4.4 besitzt die ausgewählte Zelle die Koordinaten 5 und 5, das bedeutet der zweite Ton des Dreiklangs ist fünf Halbschritte über dem Grundton, und der dritte Ton ist zehn Halbtonschritte über dem Grundton. Die zugehörige Orchesterzelle befindet sich im Zustand vier und es erklingt der Dreiklang mit dem zuvor definierten Instrument. Die Nachbarzellen bestimmen die Dauer der Klänge.

Anmerkungen zu Camus

Bei Camus werden Tonverhältnisse durch Koordinatenmapping einer Zelle bestimmt. Das Problem bei dieser Art des Mapping stellt unter anderem die Festlegung des Koordinatenursprungs in einer Ecke des Gitters dar. Bei einem 2-dimensionalen Automaten gibt es in diesem Sinne keinen Ursprung, weder in x - noch in y -Richtung. Nimmt eine Zelle einen bestimmten Wert an, geschieht das aufgrund der Nachbarschaftsverhältnisse und der definierten Regeln, unabhängig von einem willkürlich definierten Ursprungs. Es werden der Abstand vom Ursprung sowohl auf der x -Achse als auch auf der y -Achse zur Bestimmung von Intervallen verwendet. Dies repräsentiert aber die Entwicklung einer Zelle nicht.

Auch die Auswahl der Orchestrierung ist sehr problematisch. Das zweite Gitter ist ein eigenständiger Automat, der inhaltlich nichts mit dem ersten zur Tonhöhengenerierung gemein hat. Zudem wird wieder ein Koordinatenursprung definiert, der mit dem oberen Gitter korrespondiert. Entsteht nun eine aktive Zelle im zweiten System, so hat diese aber nichts mit der Position des ersten Gitters gemein, wird dennoch zur Auswahl eines Instrumentes herangezogen. Hier ist wieder die völlige Willkür bedenklich.

Mit Camus können zwar musikalische Strukturen generiert werden, jedoch besteht kein Zusammenhang zum verwendeten System. Dies führt zum Problem, dass eine gewünschte musikalische Struktur nicht direkt durch eine bestimmte Regel bzw. einen bestimmten ZA umgesetzt werden kann.

4.5.2 Chaosynth

Chaosynth ist ein auf einem ZA basierender Grain-Generator ². Der Name *Chaosynth* setzt sich zusammen aus dem Wort *ChaOs* - ein Acronym für Chemical Oscillator - und *Synth* - eine Anspielung auf Granular Synthesis. ChaOs besteht aus einem Array von Schaltkreisen, den sogenannten Nervenzellen. Diese Zellen können drei unterschiedliche Zustände annehmen abhängig vom Zustand ihrer acht Nachbarzellen und der Spannung, die zwischen ihnen fließt. V_{min} und V_{max} stellen Grenzwerte dar, die angeben, wann eine Zelle in einen anderen Zustand übergehen kann. Wenn die interne Spannung V_i unter V_{min} ist, befindet sich die Zelle in einem Ruhezustand. Ist V_i zwischen V_{min} (inklusive) und V_{max} , dann wird die Zelle depolarisiert. Sobald V_i den Wert V_{max} erreicht, verbrennt die Zelle zum Zeitpunkt t und wird zu einer Zelle im Ruhezustand zum Zeitpunkt $t+1$.

Der Zelluläre Automat in Chaosynth

Die einzelnen Zellen des Automaten stellen Nervenzellen dar. Das heißt, eine Zelle ist entweder bewegungslos, depolarisiert oder verbrannt. Zu jedem Zustand gibt es eine Übergangsfunktion:

- *bewegungslos*: abhängig von den Nachbarzuständen (wie viele depolarisiert oder verbrannt sind) wird die Zelle zum Zeitpunkt $t+1$ auch polarisiert oder bleibt bewegungslos
- *depolarisiert*: mit jedem Zeitpunkt $t+1$ wird die Zelle depolarisierter
- *verbrannt*: zum Zeitpunkt $t+1$ wird die Zelle bewegungslos

Das Mapping in Chaosynth

Der ZA steuert die Frequenz und die Dauer jedes Grains, wobei die Amplitude der Grains vom Benutzer selbst festgelegt wird. Die Werte der Nervenzelle bestimmen die Frequenz und Oszillatoren werden einer bestimmten Anzahl Nervenzellen zugeordnet. Die Frequenz eines Grains zu einem Zeitpunkt t wird durch das arithmetische Mittel der Frequenzen bestimmt, die auf die Nervenzellen gemappt sind.

Der Benutzer kann die Gittergröße, die Anzahl der Oszillatoren, die Anzahl der zugewiesenen Zellen zu einem Oszillator, die Frequenzen, den Widerstand und die Kapazität verändern.

²Ein Grain ist ein Audiosapmle mit einer sehr kurzen Dauer. Diese liegt üblicherweise unter 50 ms.

Anmerkungen zu Chaosynth

Bei Chaosynth entstehen lokale, oszillierende Spiralmuster. Um ein sinnvolles Mapping zu ermöglichen, müssten die jeweiligen Blöcke, die einen Oszillator steuern, sinnvoller Weise ein Spiralmuster abdecken. So könnte die Entwicklung dieser Spirale mitverfolgt werden. Dazu muss aber die Entwicklung des Musters bekannt sein. Innerhalb eines Blockes werden jedoch die Zustände der einzelnen Zellen aufaddiert, arithmetisch gemittelt und dann als Steuerparameter weiterverwendet. Meiner Meinung nach geht in diesem Fall wieder die ganze Information der einzelnen Zellen, deren Entwicklung und Zustandsänderung, verloren. Durch die Mittelwertbildung besteht kein Bezug mehr zur einzelnen Zelle.

4.6 Peter Beyls

Since a program reflects the objectives of the artist, programming is considered a method of artistic introspection. Software is thus instrumental as a functional, materialist means allowing the active manipulation of otherwise purely conceptual constructs.
[www.beyls.org]

Der aus Belgien stammende Künstler und Informatiker Peter Beyls begann schon in den Siebzigern mit der Entwicklung von Software für den künstlerischen Bereich. Seine Veröffentlichungen umfassen unter anderem Echtzeit-System-Design, Interface-Design und die Anwendung von künstlicher Intelligenz für künstlerische Zwecke. Beyls war einer der ersten, die auf dem Gebiet der generativen Musik mit zellulären Automaten forschte.[www.beyls.org]

In [Beyls, 2004] beschreibt Beyls Überlegungen zu generellen Mapping-Strategien von ZA. Im folgenden Abschnitt sind einige seiner Ideen wiedergegeben.

- Systematische Grundprinzipien um ZA als Klänge zu interpretieren:
 - static system:** *discrete, arbitrary linear array functioning as lookup table;*
 - dynamic system:** *continuous, an invented procedural mapper of arbitrary complexity;*
 - living systems:** *a human performer, non linear;*
 - evolved cultural system:** *e.g. the concept of tonality;*

psycho-acoustic system: for instance, dissonance and its resolution;

natural systems: laws of form and proportion i.e. golden section.

- Um Randbedingung und Regeln ableiten zu können, muss ein direkter Zusammenhang zum System herstellbar sein. Als Beispiel nennt Beyls die Generierung von Tonhöhe oder Tondauer direkt aus der XY-Position und dem Zustand der Zelle des ZA. Es muss ein numerischer Zusammenhang geschaffen werden, auch wenn dieser erst vom User definiert wird.

A fundamental cultural choice will thus steer the result.

- Er definiert den Grad der Qualität des Mappingprozesses über die Nachvollziehbarkeit. Sobald der/die AnwerderIn den Zusammenhang nicht mehr erkennt, das der Mappingalgorithmus zu komplex wird, führt es zu einem Kontrollverlust.

An important conclusion is that complexity increases interestingnes and decreases predictability/usability of any mapping scheme.

Anmerkungen zu Peter Beyls

Meiner Meinung nach macht es Sinn, den Grad der Qualität der Mappingstrategie über die Nachvollziehbarkeit zu definieren. Je komplexer ein ZA ist, desto schwieriger wird es auch, diesen musikalisch sinnvoll umzusetzen. Beim Mapping der Tonhöhe aus der XY-Position einer Zelle wird es schon schwieriger, wenn ein Koordinatensystem auf den gesamten Gitterraum angewandt wird. Die erste Frage, die sich stellt, in welchen Punkt der Ursprung gesetzt wird, ist essentiell.

Kapitel 5

Mögliche Mappinstrategien

In Kapitel 3 wurden etliche Beispiele von ZA beschrieben, so wie sie in unterschiedlichen Bereichen der Wissenschaft Anwendung finden. Nun werden daraus Anwendungsmöglichkeiten im Hinblick auf die musikalische Strukturgenese durch diese diskutiert. Es werden die Beispiele aus den Bereichen Verkehr, Chemie, Physik und Biologie herangezogen und explizite Anwendungen theoretisch erläutert.

Im Folgenden wird stets der Begriff *Klang* für jegliche Art von musikalischem Einzelereignis angewandt. Der *Klang* kann als, Ton, Akkord, Cluster, Sinuston, Geräusch etc. interpretiert werden. Diese Vereinfachung wird aus dem Grund gemacht, da der Begriff der musikalischen Strukturen sehr weitläufig ist und im Fall der ZA auf jegliche Art von Klang angewendet werden kann.

5.1 Horizontale Denkweise

In der bisherigen Anwendung von ZA zur musikalischen Strukturgenese wurden 1-dimensionalen Zellräumen nicht selten horizontal ansteigende Tonhöhen zugewiesen. Wie zuvor erwähnt, gibt es bei ZA jedoch keine logische Orientierung von links nach rechts oder umgekehrt, da alle Zellen gleichberechtigt sind. Dieser Umstand wird beim *NaSch-Modell* anders interpretiert. In diesem Fall bleibt die Anzahl der aktiven Zellen im Gitter immer gleich, durch das zugrundeliegende Regelwerk wird eine Bewegung der aktiven Zellen von links nach rechts erzwungen. In diesem Fall wird es legitim, horizontal zu denken. Verfolgt wird nicht die Entwicklung der selben Zelle vertikal in der Zeit, sondern des aktiven Zustands in der horizontalen Ebene. Dadurch, dass die aktiven Zellen

Autos mit unterschiedlichen Geschwindigkeiten simulieren, kann in diesem Fall das Auto (also die aktive Zelle) an sich musikalisch interpretiert werden. Konkret werden dabei die unterschiedlichen Zustände als musikalische Parameter definiert.

Hier genannt sei die Generierung rhythmischer Strukturen mittels des zuvor genannten Algorithmus. Dabei stellt der Zustand 0 eine Pause dar, der Zustand 1 eine ganze Note, der Zustand 2 eine halbe Note, der Zustand 3 eine Triole usw.. Geht man einen Schritt weiter, können die einzelnen Zustände auch schon zuvor rhythmisch auskomponierte Takte sein. Jedes Auto stellt dabei ein Instrument bzw. einen Klang dar.

5.2 Betrachtung der Symmetrie

Wird ein 1-dimensionaler ZA mit dem Radius $r=1$ und zwei möglichen Zuständen $k=2$ in der Zeit dargestellt, mit der Anfangskonfiguration dass nur eine Zelle in der Mitte des Zellraums den Zustand 1 besitzt, ergibt sich für bestimmte Regeln (z.B. Regel 126, 133, 150) sofort ein symmetrisches Muster. Die Frage ist, legitimiert ein symmetrisches Muster, dessen Ursprung in der Mitte liegt, die Betrachtung des ZA mit einem Ursprung?

5.3 Dynamik von Klängen

Im Bereich *Verkehr* wurden das *NaSCH-Modell*, das *BML-Modell* und das *CS-Modell* vorgestellt. Sie dienen dazu, den Verkehrsfluss und Phänomene wie Staubbildung zu simulieren.

Das *NaSCH-Modell* und das *BML-Modell* bieten eine Möglichkeit, die Tempi einzelner Klänge zu definieren. Konkret stellen dabei die jeweiligen Zustände unterschiedliche Tempi dar. Ein Auto wäre in diesem Fall ein Klang. Ein Klang kann *beschleunigt* oder *verlangsamt* werden, ebenso ist es möglich, ein Tempo zu halten, wenn die Dichte der Klänge einen Schwellwert nicht überschreitet. Somit stellen die zuvor erwähnten Modelle Möglichkeiten dar, *accelerando* und *ritardando* auszukomponieren. Das Gleiche gilt auch für die Lautstärke von Klängen oder sogar musikalischen Abschnitten. Es können Dynamiken wie *crescendo* und *decrescendo* oder *diminuendo* erzeugt werden.

Weiters stellt das *Modell der Misch-Masch-Maschine* eine Möglichkeit zur Generie-

rung von jeglicher Art von Klangdynamiken dar. Bei diesem Modell gibt es *gesunde*, *infizierte* und *kranke* Zellen. Die zwei Extremwerte *gesund* und *krank* stellen äußere Grenzwertpaare wie *laut & leise*, *hoch & tief*, *schnell & langsam*, *links & rechts*, *hell & dunkel* usw. dar. Der Grad der Infektion gibt die Zwischenwerte als Annäherung an. Somit können Dynamiken jeglicher Art generiert werden.

5.4 Der Nicht-Klang - eine Pause

Ebenso wichtig für eine Komposition wie der Klang, ist der *Nicht-Klang*, damit gemeint ist eine Pause, eine Zäsur, eine Unterbrechung während des Stückes. Für die Generierung von Pausen stellt das *CS-Modell* eine sehr gute Methode dar. Hauptsächlich dient es zur Simulation der Wechselwirkung von Fahrzeugen untereinander sowie der Fahrzeuge mit den Ampeln. Schaltet eine Ampel auf rot, wird die Geschwindigkeit des Fahrzeuges vermindert bis es zum Stillstand kommt. Stellt das Fahrzeug einen Klang dar, kann der Stillstand bei der roten Ampel aus *Pause* interpretiert werden.

Das *NaSch-Modell* und das *BML-Modell* können ebenfalls Pausen in Form einer Staubildung simulieren. Beim *NaSch-Modell* kann das Fahrzeug, sobald es den Wert 0 erreicht und es zum Stillstand kommt, als Pause gesehen werden. In diesem Fall wäre der Stillstand ein zusätzlicher Extremwert hinsichtlich der Betrachtung der aktiven Zelle als Klangdynamikgenerator.

5.5 Bewegung der Klänge

Um die Bewegungsrichtung eines Klanges zu definieren, kann das Modell der *Zellularen Gittergase* herangezogen werden. Als konkretes Beispiel wird hier die Wiedergabe der Komposition mit einem Ambisonics-System¹ beschrieben. Ein 2-dimensionale Gitter des Modells stellt in diesem Fall eine Ebene der Ambisonics-Sphäre dar. Um eine dreidimensionale Bewegungsrichtung zu erreichen, können zwei Gittergasmodelle parallel verwendet werden. Die einzelnen Zellen bzw. Gitterpunkte können einmal als x- und

¹Das Ambisonics-System stellt ein Verfahren zur Aufnahme und Wiedergabe eines Klangfeldes dar. Dabei liegt das Hauptaugenmerk auf der Reproduktion des Schallfeldes im dreidimensionalen Raum. Realisiert wird die Aufnahme mit mehreren Mikrofonen mit unterschiedlichen Richtcharakteristiken an einer Stelle. Wiedergegeben wird das Signal über etliche Lautsprecher in einem Raum. Die Klangsphäre unter dieser Lautsprecheranordnung wird als Ambisonics-Sphäre bezeichnet. [Sontacchi Höldrich]

y-Koordinaten betrachtet werden, beim zweiten Gittermodell als x- und z-Koordinate. Ein Teilchen im Gittergasmodell stellt dabei einen Klang dar.

Das *bionische ZA-Modell* dient zur Simulation von Fußgängerdynamiken. Im speziellen werden damit Phänome wie Staubbildung, Bildung von Trampelpfaden, Oszillationen und Flussmuster dargestellt. Wie schon zuvor erwähnt, können mit dem *statischen Bodenfeld* Geometrieeffekte im 2-dimensionalen Raum sehr gut simuliert werden. Dies stellt ebenso eine Möglichkeit dar, die Bewegung einzelner Klänge im Raum zu positionieren, vor allem ab einer quadrophonischen Wiedergabesituation über Lautsprecher. Die einzelnen Fußgänger, d. h. aktive Zellen, stellen einen Klang dar. Dieser wird in Abhängigkeit von Bodenfeld im Raum bewegt. Ebenso kann die Präferenzmatrix mit unterschiedlichen Gewichtungen die bevorzugte Bewegungsrichtung beeinflussen.

5.6 Klangdichte

Das Modell des *Planeten Wator* beschreibt ein Ökosystem von Fischen und Haien. Zur Generierung musikalischer Strukturen kann es sowohl auf der Mikroebene als auch auf der Makroebene der Klangproduktion zum Einsatz kommen. Die Mikrostruktur beschreibt einen einzelnen Klang an sich, seine Dichte; die Makrostruktur beschreibt die Klangdichte des gesamten Werkes. Dabei stellen Fische und Haie entweder einzelne Klangkomponenten oder gesamthafte Klänge dar. Die Reproduktion der Tiere bewirkt dabei eine Zunahme der Klangdichte. Ebenso besteht die Möglichkeit, dass ein Klang oder eine Komponente verloren gehen, wenn der Fall eintritt, dass entweder Fische oder Haie aufgrund von Nahrungsmangel aussterben.

5.7 Ausblick

Finally, a CA is also a system which favors serendipity i.e. finding interesting patterns without looking for them explicitly. After all, there is no single best solution; the user navigates areas of relative interestingness and probes a given area to retrieve exemplary material. [Beyls, 2004]

Schlussendlich kann gesagt werden, dass die Anwendung zellulärer Automaten im

Hinblick auf musikalische Strukturgenese eine Vielzahl von Möglichkeiten für die Komponistin darstellt. Jedoch gibt es nicht, wie schon Peter Beyls sagte, eine explizite Lösung für ein Problem, sondern die Anwenderin tastet sich durch ein Feld von zahlreichen Mustern und Regeln, um dann durch Trial and Error zum gewünschten musikalischen Material zu gelangen.

Literaturverzeichnis

- [Beyls, 1989] Beyls, P.: *The musical universe of cellular automata*, In Wells, T. and Butler, D. (eds) Proceedings of the 1989 International Computer Music Conference. International Computer Music Association, S. 34–41.
- [Beyls, 2003] Beyls, P. : *Selectionist musical automata : Integrating explicit instruction and evolutionary algorithms*, IX Brazilian Symposium on Computer Music. Brazilian Computing Society, 2003
- [Beyls, 2004] Beyls, P. : *Cellular automata mapping procedures*, Proceedings of the 2004 International Computer Music Conference
- [Bilotta Pantano, 2001] Bilotta, E.; Pantano, P.: *Artificial Life Music Tells of Complexity*, Almma 2001: Proceedings of the workshop on artificial life models for musical applications. Linguistics Department, University of Calabria, Arcavacata di Rende, 2001
- [Bilotta Pantano , 2002] Bilotta, E., Pantano, P. : *Synthetic Harmonies: An Approach to Musical Semiosis by Means of Cellular Automata*, LEONARDO, vol. 35, S. 35-42, MIT Press, 2002
- [Bilotta Pantano Talarico, 2000] Bilotta, E.; Pantano, P.; Talarico, V.: *Music Generation through Cellular Automata: How to give Life to Strange Creatures*, Proceedings of Generative Art GA 2000, Mailand, Italien
- [Burks, 1970] Burks, A.: *Essays on Cellular Automata*, University of Illinois Press, 2000
- [Burraston Edmonds, 2005] Burraston, D.; Edmonds, E.: *Cellular Automata in Generative Electronic Music and Sonic Art : A Historical and Technical Review*, Digital Creativity 16(3) S.165-185

- [Dorin, 2002] Dirubm A.: *Liquiprism: Generating Polyrythms with Cellular Automata*, Proceedings of the 8th International Conference on Auditory Display, S. 447-451, 2002
- [Eigen, 1975] Eigen, M., Winkler, R. (1975): *Das Spiel. Naturgesetze steuern den Zufall.*, Piper, München und Zürich., 1975
- [Gerhardt Schuster, 1995] Gerhardt, M.; Schuster, H.: *Das digitale Universum Zelluläre Automaten als Modelle der Natur*, Fried. Vieweg & Sohn, 1995
- [Goos, 1998] Goss, G.: *Vorlesungen über Informatik: Band 4: Paralleles Rechnen und Nicht-Analytische Lösungsverfahren*, Springer, 1998
- [Hunt Kirk Orton, 1990] Hunt, A.; Kirk, R.; Orton, R. : *MidiGrid: An Innocative Computer-Based Performance and Composition System*, ICMC Glasgow 1990 Proceedings
- [Hunt Kirk Orton, 1991] Hunt, A.; Kirk, R.; Orton, R. : *Musical applications of a cellular automata workstation*, Proceedings of the 1991 International Computer Music Conference. Montreal, Canada, S.165–168.
- [Ilachinsky, 2001] Ilachinsky, A.: *Cellular Automata A discrete Universe*, World Scientific Publishing Co. Pte. Ltd., 2001
- [Millen, 1990] Millen, D.: *Cellular Automata Music*, Proceedings of the 1990 International Computer Music Conference. ICMA, San Francisco, S.314–316.
- [Millen, 1992] Millen, D.: *Generation of formal patterns for music composition by means of cellular automata*, Proceedings of the 1992 International Computer Music Conference. San Francisco: ICMA, S.398–399.
- [Millen, 2004] Millen, D.: *An Interactive Cellular Automaton Music Application in Cocoa*, Proceedings ICMC 2004
- [Miranda, 1998] Miranda, E.: *Computer Sound Design: Synthesis Techniques and Programming*, Focal Press Oxford
- [Miranda, 2001] Miranda, E.: *Evolving Cellular Automata Music: From sound synthesis to composition*, Almma 2001: Proceedings of the workshop on artificial

- life models for musical applications. Linguistics Department, University of Calabria, Arcavacata di Rende, 2001
- [Miranda, 2004] Miranda, E.: *At the Crossroads of Evolutionary Computation and Music: Self-Programming Synthesizers, Swarm Orchestras and the Origin of Melody*, MIT, Evolutionary Computation 12 (2), S. 137-158
- [Miranda, 2005] Miranda, E.: *On making Music with Artificial Life Models*, In Engineering Nature: Art and Consciousness in the Post-Biological Era, R. Ascott (Ed.). Bristol, UK: Intellect
- [Nagel Schreckenberg, 1992] Nagel, K.; Schreckenberg, M.: *A cellular automata model for freeway traffic* J. Phys. I France 2, S. 2221-2229, 1992
- [Nierhaus, 2009] Nierhaus, G.: *Algorithmic Composition, Paradigms of Automated Music Generation*, Springer Verlag Wien, 2009
- [Schadschneider, 2004] Schadschneider, A.: *Physik des Straßenverkehrs* Skript zur gleichnamigen Vorlesung Version 23. April 2004, Universität Köln
- [Sontacchi Höldrich] Sontacchi, A.; Höldrich, R.: *Schallfeldreproduktion durch ein verbessertes Holophonie - Ambisonics System*
- [Stegemann, 2001] Stegemann, T.: *Erfolgreiche Chaoskontrolle*
www.heise.de/tp/artikel/9/9794/1.html
- [Toffoli Margolus, 1985] Toffoli, T.; Margolus, M.: *Cellular Automata Machines A new environment for modeling*, MIT Press Series in Scientific Computation, 1985
- [Turing, 1952] Turing, A.: *The chemical basis of morphogenesis*, Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences, Vol. 237, No. 641. (Aug. 14, 1952), pp. 37-72
- [Von Neumann, Burks, 1966] Von Neumann, J.; Burks, A.: *The theory of Self-Reproducing Automata*, University of Illinois Press, Urbana and London, 1966
- [Wolfram, 2002] Wolfram, S.: *A New Kind of Science*, Wolfram Media Inc., 2002

[Wolfram, 2011] Wolfram, S.: *Music, Mathematica and the Computational Universe*,
www.stephenwolfram.com, 2011

[Wolfram, 2012] Wolfram, S.: *How WolframTomes Works*, www.stephenwolfram.com,
2012

[Zuse, 1969] Zuse, K.: *Rechnender Raum, Schriften zur Datenverarbeitung Band I*,
Friedr. Vieweg & Sohn GmbH, Braunschweig, 1969

[www.abovetopsecret.com]

[www.beyls.org]

[www.heise.de]

[www.midigrid.fullpitcher.co.uk/index.html]

[www.stephenwolfram.com]

[www.wikipedia.org]

Abbildungsverzeichnis

1.1	Bsp. für einen 2-dim. Zellraum mit $n \times m$ Zellen	7
1.2	a) Moore-Nachbarschaft b) von Neumann-Nachbarschaft im 2-dim. Zellraum	8
1.3	a)periodische b) symmterische Randbedingung für 1-dim. Automaten . .	8
1.4	Toriod [www.abovetopsecret.com]	9
1.5	4 Klassen des 1-dim Automaten www.stephenwolfram.com	10
1.6	3-dim. Automaten mit 6 und 26 Nachbarzellen S. 182f [Wolfram, 2002] .	13
2.1	a) Blinker und b) Block in GoL	22
2.2	Entwicklung einer simplen Startkonfiguration	22
2.3	Glider Gun	23
2.4	Auslöschung	23
2.5	r-Pentominog	24
2.6	Fresser	24
2.7	1-dim. Automaten	26
2.8	Komplexität in Abhängigkeit des Langton-Parameters	29
3.1	Verkehrssimulation mit Staubildung [<i>Nagel Schreckenberg, 1992</i>]	32
3.2	BML-Modell ohne Staubildung http://www.wikipedia.org	33
3.3	BML-Modell mit Staubildung http://www.math.ubc.ca/~holroyd/bml/ . .	33
3.4	CS-Modell: s, d [<i>Schadschneider, 2004</i>]	35
3.5	CS-Modell mit Gridlock [<i>Schadschneider, 2004</i>]	35
3.6	Phänomen der Staubildung [<i>Schadschneider, 2004</i>]	36
3.7	Entstehung eines Kreisverkehrs [<i>Schadschneider, 2004</i>]	37
3.8	Statisches Bodenfeld [<i>Schadschneider, 2004</i>]	38
3.9	Präferenzmatrix [<i>Schadschneider, 2004</i>]	38
3.10	Misch-Masch-Maschine www.schockwellenreiter.de	41
3.11	Muster während einer CO-Oxidation www.heise.de	42

3.12	Blöcke der Margolus-Nachbarschaft [<i>Goos, 1998</i>]	43
3.13	Bewegungsrichtung durch Blockregel	43
3.14	Zusammenstoß von zwei Teilchen im HPP-Gas [<i>Goos, 1998</i>]	44
3.15	Entwicklungsregeln im HPP-Gas	44
3.16	Typische Muster in AIM [<i>www.stephenwolfram.com</i>]	45
3.17	Entwicklungsschleife und Musterbildung des Hyperzyklus	47
3.18	Simulation Planet Wator	50
4.1	Mapping in BOG [<i>Bilotta Pantano, 2001</i>]	54
4.2	Mapping eines Akkordes in Camus [<i>Miranda, 2001</i>]	62
4.3	Mapping von aktiven Zellen in Camus [<i>Miranda, 2001</i>]	62
4.4	Klangerzeugung in Camus [<i>Miranda, 2001</i>]	62