

2 Zelluläre Automaten – Ein ausgewählter Überblick

Uwe Renner*

2.1 Einleitung

Zelluläre Automaten (ZA) erlangen aufgrund ihrer vielseitigen Anwendbarkeit zunehmende Beachtung in den verschiedensten Wissensbereichen, so in der Physik, in der Chemie und in der Biologie, aber auch in der Medizin, in der Ökologie und in der Wirtschaft. Als *Simulationsmodell* ermöglichen sie es, die hier vorkommenden komplexen Abhängigkeiten in ihrer zeitlichen und räumliche Entwicklung zu untersuchen.

Die Modellkonzeption des zellulären Automaten besteht vereinfacht darin, die Eigenschaften von Einzelobjekten, d. h. den Zellen, als Zustände zu formulieren und für deren zeitliche Entwicklung *Regeln* aufzustellen, welche diese Zustände wieder in Zustände aus einer vorgegebenen endlichen Auswahl transformieren, wobei zusätzlich die Zustände der benachbarten Zellen berücksichtigt werden. Die Regeln können streng aus bekannten Gesetzen abgeleitet, aber auch als Erfahrungsregeln formuliert werden.

Zelluläre Automaten erlauben das Rechnen mit vagen Formulierungen und entsprechen damit den Grundvorstellungen des *Soft Computing*. Obwohl mit zellulären Automaten auch numerisch gerechnet werden kann, liegt bei ihnen jedoch der Schwerpunkt in der Formulierung von Überführungsregeln für die Zustände. Quantitative Aussagen ergeben sich dann aus der Interpretation dieser Zustände, so etwa aus ihrer statistischen Verteilung oder den räumlichen bzw. zeitlichen Mittelwerten.

Als *Hardwarerealisierung* stellen zelluläre Automaten aufgrund ihrer massiv-parallelen Arbeitsweise ein Alternative zu den bestehenden Architekturen dar,

*Arnold-Sommerfeld-Gesellschaft e.V., Thaerstraße 34, D-04129 Leipzig

so vor allem zu den herkömmlichen sequentiell arbeitenden Computern, aber auch zu vielen Parallelrechnern.

2.2 Historische Vorbemerkungen

Bei der Entstehung der Modellkonzeption des zellulären Automaten haben viele namhafte Wissenschaftler mitgewirkt. Vor allem ist hier JOHN VON NEUMANN zu nennen, der sich von 1948 bis zu seinem Tode im Jahre 1957 mit der *Architektur* von Computern und der Entwicklung universeller *Algorithmen* befaßte. Seine Beiträge zur Geschichte der Computerentwicklung sind eng mit der Entwicklung des zellulären Automatenmodells verknüpft.¹ Hier ließ er sich von der Natur inspirieren und zog Parallelen zwischen künstlichen und natürlichen Systemen. In den Vorstellungen zum zellulären Automaten unterstützte ihn maßgeblich der Mathematiker STANISLAW ULAM. Die Ergebnisse wurden von ARTHUR W. BURKS im Buch „Theory of Self-Reproducing Automata“ zusammenfassend aufgeschrieben und vervollständigt [1].

Eine der Ursachen, weshalb sich VON NEUMANN mit solchen Fragestellungen befaßte, begründet sich im Fehlen geeigneter Verfahren bei der Gewinnung quantitativer Informationen aus nichtlinearen partiellen Differentialgleichungen. Insbesondere traf und trifft dies wohl auch noch heute noch auf die Fluid-dynamik zu. So waren schon damals praktische Probleme in Zusammenhang mit Turbulenz und Schockwellen von Interesse. Für ihn stellte sich heraus, daß sich grundlegende Aussagen vorerst nur numerisch gewinnen ließen. Die Ergebnisse könnten später wiederum von theoretischem Nutzen sein und auch experimentell geprüft werden. Ein solches Vorgehen entspricht zugleich der traditionellen hypothetisch-deduktiven Methode bei der Findung von Erkenntnissen, wonach die gesuchte Information in einer iterativen Folge gewonnen wird. Für VON NEUMANN war der Computer vor allem ein *heuristisches Werkzeug*. Erst im kombinierten Einsatz mit dem realen Experiment sah er in ihm zugleich ein Simulationswerkzeug.

VON NEUMANN entwarf zahlreiche Computerkonzepte, von denen die *nach ihm* benannte Architektur mit den Bestandteilen zentrale Steuereinheit **CC** (heute CPU: central processing unit), zentrale Arithmetikeinheit **CA** (heute ALU: arithmetic logical unit), interner und externer Speicher (ursprünglich in besonders schnellen Speicher **M** und externes Aufzeichnungsmedium **R** getrennt), Eingabeeinheit **I** und Ausgabeeinheit **O** die wohl bekannteste

¹Es sei angemerkt, daß schon 1941 KONRAD ZUSE einen programmgesteuerten Rechner mit Binärzahlendarstellung und Gleitkommaarithmetik unter der Bezeichnung Z3 baute.

ist. Dabei war eines seiner Entwicklungsziele der „Allzweckrechner“ (general purpose computer). Auch war er maßgeblich bei der Verbesserung des legendären ENIAC² beteiligt. Der Einsatz schnellerer und zuverlässigerer Speicherelemente mit hoher Kapazität und die Programmierbarkeit der Computer (Einführung des Programmspeichers, Flußdiagramme, Programmiersprachen) waren einige seiner Vorschläge. Hinsichtlich der Entwicklung möglicher Programmiersprachen sah er diese vor allem unter dem Aspekt der praktischen Anwendung von TURINGS Beweis für die Existenz eines universellen Computers (TURING-Maschine). Bei der Neukonzipierung des ENIAC-Nachfolgetyps unter der Bezeichnung EDVAC³ kamen viele seiner Ideen zur Realisierung.

Als Schwerpunkte beim Computorentwurf waren für VON NEUMANN die Trennung der logischen Einzelemente von der Schaltung, der Vergleich des Computers mit dem menschlichen Nervensystem, allgemeine Organisationsprinzipien sowie die Programmierung und die Steuerung wichtig. Gleichfalls standen Probleme der *Zuverlässigkeit* logischer Systeme im Mittelpunkt seiner Untersuchungen. Eine wesentliche Frage war für ihn, wie aus unzuverlässigen Komponenten zuverlässige Systeme konstruiert werden können, eine Eigenschaft, die auch viele natürliche Systeme besitzen. Als ein weiteres Merkmal wollte er die Fähigkeiten zur *Selbstdiagnose* und zur *Selbstreparatur* in seine Vorstellungen zum Computer einbringen.

Auch wollte VON NEUMANN die zahlreichen logischen Operationen in Bausteinen ähnlich dem Nervensystem räumlich organisieren,⁴ wobei er sich durch die Arbeiten von MCCULLOCH und PITTS motivieren ließ. Im Gegensatz zu den *neuronalen* Elementen sollten diese für den Einsatz im Computer den folgenden idealisierten Aufbau besitzen: drei exzitorische Eingänge, ein oder zwei inhibitorische Eingänge, Schwellwerte 1, 2, 3 und eine Verzögerungseinheit. Ein Stimulus wird mit der zeitlichen Verzögerung um eine diskrete Einheit nur unter der Bedingung ausgegeben, wenn kein inhibitorischer Eingang aktiv und die Zahl der stimulierten exzitorischen Inputs wenigstens so groß wie der Schwellwert ist. Durch Einführung solcher Funktionselemente wird die Entwicklung der Computerlogik *unabhängig* vom jeweiligen Stand der Technologie.

Die so getroffenen Idealisierungen waren auch der Ausgangspunkt für die weitere Beschäftigung VON NEUMANNs mit der *Automatentheorie*, welche er als die allgemeine Computertheorie ansah, die Strukturkonzepte und Organisationsprinzipien sowohl natürlicher als auch künstlicher Systeme, deren Sprache

²Electronic Numerical Integrator and Computer

³Electronic Discrete Variable Automatic Computer

⁴Heutige programmierbare Logikschaltungen, insbesondere FPGA (Field Programmable Gate Array), entsprechen dieser Vorstellung weitgehend.

und Information sowie ihre Programmierung und Steuerung umfaßt. Dabei besitzt die *mathematische Logik* einen gewichtigen Anteil. VON NEUMANN zog Parallelen zwischen der Automatentheorie, der Informationstheorie und der Thermodynamik.

Seine Bestrebungen, auch *lebensähnliche* Strukturen mit logischen Methoden zu beschreiben, fanden ihren Höhepunkt in der Darstellung eines *selbstreproduzierenden* Automaten, dessen Zellen sich in 29 Zuständen befinden können ([1], Teil II, Kapitel 2). Der so konstruierte Automat besaß die Eigenschaft der *Universalität* einer TURING-Maschine und konnte sich in seiner Umgebung selbstreproduzierend entfalten, was Grundlage einer evolutionären Entwicklung ist. Er war auch in der Lage, seine Regeln weiterzugeben, ähnlich einem richtigen Lebewesen. Diese Vorstellungen ähneln stark den heutigen über den genetischen Vererbungsprozeß. VON NEUMANN'S Ideen zur Selbstreproduktion wurden später von CODD und LANGTON weiterentwickelt und in der Zustandszahl – auf acht – stark reduziert. Im letzteren Fall wurde aber die Forderung nach der Universalität zugunsten der Einfachheit der Regeln wieder aufgegeben.

Ein weiterer Meilenstein in der Entwicklung des zellulären Automaten wurde durch JOHN HORTON CONWAYS mit LIFE im Jahre 1968 gesetzt. Dieses Spiel fand in den darauffolgenden Jahren populäre Verbreitung, die heute noch anhält. STEPHEN WOLFRAM untersuchte viele neue Anwendungsfelder zellulärer Automaten, so insbesondere in der Physik. Ihm gelang auch eine Typisierung der eindimensionalen zellulären Automaten [2]. Gleichsam wurden diese zum Gegenstand der *Komplexitätsforschung*. Weiterhin trugen EDWARD FREDKIN, TOMMASO TOFFOLI und NORMAN MARGOLUS maßgeblich zum heutigen Verständnis der zellulären Automaten bei.

2.3 Grundlegende Eigenschaften

Der Zelluläre Automat $\mathcal{A} = (\mathcal{G}, \mathcal{Z}, \mathcal{N}, F)$ wird definiert durch die (reguläre) Gittergeometrie \mathcal{G} , in welcher die einzelnen Zellen (Einzelautomaten) im Einbettungsraum angeordnet sind, die endliche Menge diskreter Zustände \mathcal{Z} , auch Alphabet genannt, die lokale und uniforme Nachbarschaft \mathcal{N} und eine für jede der Zellen geltende Überföhrungsfunktion F , die den Zustand einer Zelle in Abhängigkeit von den Zuständen der Nachbarschaftszellen in diskreten Zeitschritten wieder in einen Zustand der Zustandsmenge \mathcal{Z} transformiert. Der Zustand des zellulären Automaten ändert sich räumlich parallel und synchron mit dem globalen Zeittakt. Die zeitliche Entwicklung des zellulären Automaten wird maßgeblich durch die Anfangskonfiguration der Zellen festgelegt.

Zahlreiche Erweiterungen dieser Definition sind denkbar. Da die Automatenzelle nach obiger Definition nur ein „Kurzzeitgedächtnis“ besitzt, wird nur der aktuelle Zustand transformiert. Die Berücksichtigung weiter zurückliegender Zustände wäre jedoch ohne weiteres möglich. Teilweise lassen sich die Einzelautomaten zu Automatenkomplexen organisieren oder auch in Subautomaten zerlegen.

2.3.1 Gittergeometrien

Die einzelnen Zellen sind in einem euklidischen Einbettungsraum miteinander verbunden und bilden einen räumlichen Komplex von meist *homogener* und *regulärer* Struktur, dessen Größe bzw. Zellenanzahl aus praktischen Gründen, wie der zur Verfügung stehenden Speicherkapazität, jedoch beschränkt ist.

Der einfachste Fall ist die lineare Anordnung der Zellen in einer Kette. Zur Darstellung der zeitlichen Entwicklung des Zustandes kann die Zeit als weitere Dimension hinzugenommen werden, so daß ein räumlich-zeitliches Bild entsteht. Lineare eindimensionale zelluläre Automaten mit zwei Zuständen gelten in ihrer Entwicklung als wohlverstanden.

Bei den *zweidimensionalen* Geometrien sind vor allem reguläre Gitter wie das quadratische Gitter, das Dreiecksgitter und das hexagonale Gitter von Bedeutung. Dreiecksgitter besitzen in der unmittelbaren Nachbarschaft nur drei Nachbarn und lassen sich in der üblichen quadratischen Struktur nur mittelbar erzeugen. Das quadratische Gitter hingegen läßt sich am einfachsten realisieren, es besitzt aber für manche Anwendungen nur unzureichende Symmetrie. Das hexagonale Gitter wiederum besitzt die geringste Anisotropie unter den regulären zweidimensionalen Gittern. Es muß aber zur Darstellung auf das quadratische Gitter abgebildet werden, was durch Scherung des Gitters erfolgen kann. Bei den *dreidimensionalen* Gittern ist nur das kubische Gitter von praktischer Bedeutung. Die grafische zweidimensionale Darstellung erfolgt z. B. durch Schnitte. Mitunter ist man zur Erfüllung von Symmetrieforderungen auf noch höherdimensionale Gitter angewiesen, wobei man die gewünschten Eigenschaften in der interessierenden Dimension durch Projektion erhält.

2.3.2 Zellnachbarschaften

Die Zellen (i, j) stehen untereinander über die Gittergeometrie mit den Nachbarzellen in Verbindung.⁵ Von praktischer Bedeutung sind insbesondere die

⁵Es soll hier nur der zweidimensionale Fall betrachtet werden. Zellen in höheren Dimensionen lassen sich durch die entsprechende Anzahl an Indizes eindeutig kennzeichnen.

VON-NEUMANN-Nachbarschaft und die MOORE-Nachbarschaft (Abb. 2.1). Zur VON-NEUMANN-Nachbarschaft im engeren Sinne gehören neben der Zelle selbst nur die nächsten Nachbarzellen (k, l) , welche der Abstandsmetrik $|k - i| + |l - j| \leq 1$ genügen, zur erweiterten VON-NEUMANN-Nachbarschaft mit dem Radius R auch solche, die durch die Metrik $|k - i| + |l - j| \leq R$ erfaßt werden. Die MOORE-Nachbarschaft mit dem Radius R wird hingegen durch die Abstandsmetrik $|k - i| \leq R \wedge |l - j| \leq R$ definiert. Weitere Nachbarschaftsformen sind denkbar, so insbesondere stochastisch erzeugte mit unterschiedlichen Radien, wobei dieser als ein Maß für die Stärke der Wechselwirkung angesehen werden kann. Zudem kann die Nachbarschaft asymmetrisch aussehen. Die Größe der Nachbarschaft sollte aus verschiedenen Gründen, wie der zur Verfügung stehenden Simulationszeit, so klein als möglich gewählt werden, da ansonsten die Anzahl der Regeln dramatisch anwachsen würde.

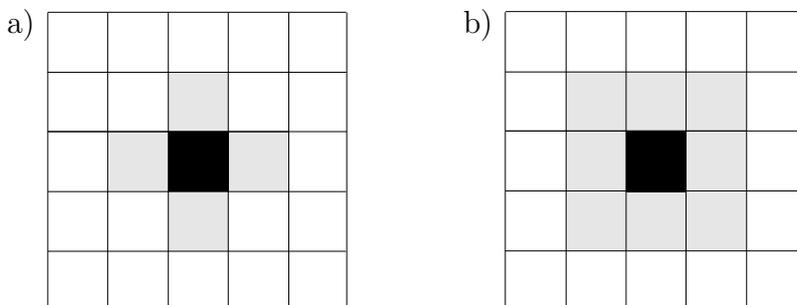


Abb. 2.1: Nachbarschaften: a) VON NEUMANN, b) MOORE

2.3.3 Zustände der Automatenzelle

Die quantitative Charakterisierung der Zellen erfolgt durch Zustände $z \in \mathcal{Z}$, deren „Werte“ o. b. d. A. ganzzahlig gewählt werden. Ihre Zustandsmenge ist *abzählbar* und sollte möglichst klein gehalten werden. Die Zustände müssen selbst keine Zahlen sein, sondern können irgendeine symbolische Bedeutung besitzen. Sie müssen unterscheidbar sein und sich durchnummerieren lassen, etwa durch z_1, z_2, \dots . Im einfachsten Falle sind sie Ja-Nein-Entscheidungen oder im konkreten Fall quantenmechanische Zustände wie der Elektronenspin. Als Zustände lassen sich auch Klassen für bestimmte statistische Merkmale verwenden, jedoch nicht die konkreten Zahlen selbst. Soll die Zelle durch mehrere unabhängige Eigenschaften gekennzeichnet werden, so lassen sich diese in einen gemeinsamen Zustand kodieren, wobei die Zustandszahl entsprechend groß zu wählen ist.

2.3.4 Lokale Regeln für die Zustandsüberführung

Grundlage für die zeitliche Entwicklung beim zellulärer Automaten sind die *Überföhrungsfunktionen* bzw. die lokalen Regeln, die lokal als Zeit-Operatoren wirken. Durch diese wird der aktuelle Zellzustand in jedem diskreten Zeitschritt aktualisiert und in Abhängigkeit von diesem und den benachbarten Zuständen in einen neuen Zustand transformiert. Diese Regeln sind nicht vordergründig als numerische Rechenvorschriften, sondern vielmehr als Abbildungen zu verstehen. Die Wahl geeigneter Automatenregeln kann auf vielfache Weise erfolgen. Die Zustandsüberführung verläuft *räumlich parallel* und *zeitlich synchron* für alle Zellen im einheitlichen Takt, der Systemzeit, beginnend mit einer geeignet gewählten Anfangskonfiguration, und entspricht somit der *massiv-parallelen* Arbeitsweise eines idealen Parallelrechners.

Die Änderung der Zustände geschieht *lokal* in jeder einzelnen Zelle unter Berücksichtigung der Zustände der Nachbarzellen. Es findet aber keine direkte Modifikation der Zustände benachbarter Zellen statt. Die beschreibbaren Wechselwirkungen des Systems beschränken sich daher auf Nahwirkung. Eine Änderung von Zuständen entfernt liegender Zellen durch die Information der betrachteten Zelle kann nur zeitverzögert mit endlicher Ausbreitungsgeschwindigkeit stattfinden.

Geht man von der möglichen kombinatorischen Vielfalt der Regeln aus, dann kann die theoretische Zahl an Überföhrungen in Abhängigkeit von der verwendeten Zustandszahl und der Zahl der Nachbarn gigantisch anwachsen, und ein Testen all dieser Fälle auf praktische Relevanz wäre ein hoffnungsloses Unterfangen: Bei einer Zustandszahl von N pro Zelle und k Nachbarn inklusive der betrachteten Automatenzelle ergeben sich potentiell N^{N^k} solcher Überföhrungsmöglichkeiten.

Allgemein wird die Änderung des Zustandes z_{ij}^t zum Zeitpunkt t in der Zelle (i, j) in den Zustand z_{ij}^{t+1} in Abhängigkeit von den benachbarten Zellen ($\mathcal{Z}^k \rightarrow \mathcal{Z}$) durch die uniforme Überföhrungsfunktion F

$$z_{ij}^{t+1} = F(\{z_{kl}^t\} \mid (k, l) \in \mathcal{N}) \quad (2.1)$$

bei gegebener Anfangskonfiguration des Systems zum Zeitpunkt $t = 0$ beschrieben. Diese Abbildung stellt eine Differenzgleichung erster Ordnung in der Zeit dar.

Eine Möglichkeit zur Beschreibung der Regeln besteht in der Verwendung von *Überföhrungstabellen* der Größe N^k . Insbesondere für große Werte N und k können diese Tabellen, falls keine Vereinfachungen getroffen werden, recht umfangreich werden.

Durch anwendungsbezogene *Symmetrien* aus der Natur (z. B. Spiegel- und Rotationsymmetrien, Ausschließungsregeln, Erhaltungssätze) läßt sich diese Zahl oft stark reduzieren und somit Simulationzeit einsparen. Speziell bieten Invarianzen bei der Zustandsüberführung bezüglich der Wahl bestimmter Zellzustände eine solche Vereinfachung. Gleichfalls führen Nachbarschaftskonfigurationen, bei denen der Zustand bei Anwendung der Regel unverändert bleibt, zu einer Reduzierung der Regelanzahl. Für die Symmetrisierung können auch einfache arithmetische und logische Operationen verwendet werden. Von besonderer praktischer Bedeutung ist hierbei die sogenannte *totalistische Regel*

$$z_{ij}^{t+1} = F\left(\sum_{(k,l) \in \mathcal{N}} z_{kl}^t\right),$$

d. h., der neue Zustand wird in Abhängigkeit von der Summe der Nachbarschaftszustände gebildet.

Oftmals werden die Regeln für die Zustandsüberführung irreversibel formuliert (z. B. LIFE, s. u.), d. h., der Ausgangszustand läßt sich nicht wieder auf umgekehrtem Wege erreichen. Möchte man Prozesse nach dem Vorbild der klassischen Teilchenphysik reversibel beschreiben, so ist dies durch ein Verfahren nach FREDKIN möglich. Die hiermit erzeugten Regeln berücksichtigen den weiter zurückliegenden Zeitpunkt $t - 1$ und sind somit von zweiter Ordnung in der Zeit.

Die Zustandsüberführung kann sowohl *deterministisch* als auch *stochastisch* oder nach *Fuzzy*-Regeln erfolgen. Stochastische Automaten sind durch eine bedingte Wahrscheinlichkeit für den Übergang der Zustände gekennzeichnet. Hierzu wird z. B. eine Zielkonfiguration ausgewürfelt und aus der aktuellen Nachbarschaftskonfiguration die Wahrscheinlichkeit für die Zustandsüberführung ermittelt. Ob diese Überführung tatsächlich erfolgt, wird mit Hilfe eines Zufallsgenerators ermittelt. Somit werden nicht nur zufällige Prozesse beschreibbar, sondern auch evolutionäre Prozesse auf der Grundlage von Mutation lassen sich simulieren. Fuzzy-Automaten erfassen in jeder Zelle die Zustände als Wahrheitswerte durch ein geeignetes Zugehörigkeitsmaß. Bei der Zustandsüberführung werden die jeweiligen Zugehörigkeitswerte für diese Zustände aktualisiert. Hierzu sind jedoch umfangreichere numerische Operationen, so z. B. die Minimum-Maximum-Bildung, erforderlich. Allgemein kann der Prozeß der Zustandsüberführung durch eine beliebigen Folge von Regeln F_1, F_2, F_3, \dots erfolgen. Eine Heterogenität in der Zustandsüberführung ließe sich durch Zustandszahlerweiterung erreichen, die dann eine bestimmte selektive Wirkung auf eine Zustandsuntermenge ausübt.

2.3.5 Anfangs- und Randbedingungen

Die globale Entwicklung des Gesamtzustandes des zellulären Automaten hängt von der Anfangskonfiguration der Zellzustände empfindlich ab. Zu ihrer Festlegung muß man oft bestimmte quantitative Bedingungen berücksichtigen. Für die Populationsentwicklung sind dies z.B. die Zahl und die räumliche Verteilung der einzelnen Individuen und ihr Zahlenverhältnis, bei der Beschreibung physikalischer Vorgänge sind dies insbesondere Erhaltungsgrößen wie die Gesamtteilchenzahl, die Gesamtenergie oder der Gesamtimpuls. Ihre Werte können deterministisch vorgegeben werden oder gehorchen einem bestimmten Verteilungsgesetz.

Zur Beschreibung der globalen Entwicklung des zellulären Automaten sind zudem geeignete Randbedingungen erforderlich – ähnlich denen, wie man sie aus der Behandlung partieller Differentialgleichung zweiter Ordnung kennt. Diese bestimmen das globale Verhalten des zellulären Automaten maßgeblich. Konstruktiv bedingt, können nur endliche Gebiete beschrieben werden. Eine mögliche unendliche Ausdehnung des Randes simuliert man durch die gedachte periodische Fortsetzung des Gebietes. Weitere Möglichkeiten zur Festlegung des Verhaltens an den Rändern sind die Wahl reflektierender Wände oder die Festschreibung der Zustandswerte. Eine Sonderbehandlung in der Simulation der Randzellen braucht nicht zu erfolgen, was im Fall periodischer Randbedingung trivial ist. Für die Erfassung allgemeiner Randbedingungen fügt man zu den vorhandenen Zuständen weitere Hilfszustände hinzu, womit dann das gewählte Randverhalten selektiv erfaßbar wird.

2.3.6 Globale Eigenschaften

Die Untersuchungen der globalen Eigenschaften des *Gesamtautomaten* sind das eigentliche Ziel der Simulation, denn sie sollen wichtige Eigenschaften des zugrunde gelegten Systems so realitätsgetreu wie möglich wiedergeben. Obwohl Fragestellungen zu deren theoretisch möglichem Verhalten in der Komplexitätstheorie untersucht werden, fehlt eine zufriedenstellende Theorie, nach welcher globale Zustände allgemein vorhergesagt werden könnten. Einziger Ausweg, dennoch Informationen über solch komplexe Systeme zu sammeln, besteht derzeit eben nur in der Simulation.

Das Spektrum möglicher Systemzustände zellulärer Automaten erstreckt sich von stationären Mustern über die verschiedensten Typen *periodischer* Strukturen bis hin zu *chaotischen* Erscheinungsformen. Eine Klassifikation wurde von WOLFRAM für den eindimensionalen Fall erbracht [2]. Es lassen

sich vier *Universalitätsklassen* unterscheiden. Hiernach führen zufällig gewählte Anfangskonfigurationen zu:

1. einem homogenen Zustand,
2. einfachen, voneinander getrennten, periodischen Strukturen,
3. chaotischen Mustern,
4. komplexen lokalisierten Strukturen.

Als ein Beispiel für die komplexe Entwicklungsmöglichkeit sei die Herausbildung einer *fraktalen* Raum-Zeit-Struktur in Gestalt des SIERPIŃSKI-Fraktals genannt (s. Abb. 2.2). Die räumliche Besetzung der Zustände erfolgt entlang einer eindimensionalen Zellanordnung und kann durch die binären Zustandswerte z_i^t in diskreter zeitlicher Folge in t beschrieben werden. Ist eine Raumzelle i besetzt, so wird ihr der Wert 1 zugewiesen, andernfalls der Wert 0. Die Zustandsüberführung erfolgt mittels einer einfachen logischen Verknüpfung, dem Exklusiv-Oder (XOR), entsprechend der Formel

$$z_i^{t+1} = z_{i-1}^t \text{ XOR } z_{i+1}^t$$

und kann zeilenweise dargestellt werden, so daß sich ein zweidimensionales Bild ergibt. Die Zustände besitzen sowohl nach links als auch nach rechts eine maximale Geschwindigkeit, mit der sie sich in einem Raum-Zeit-Kegel ausbreiten.

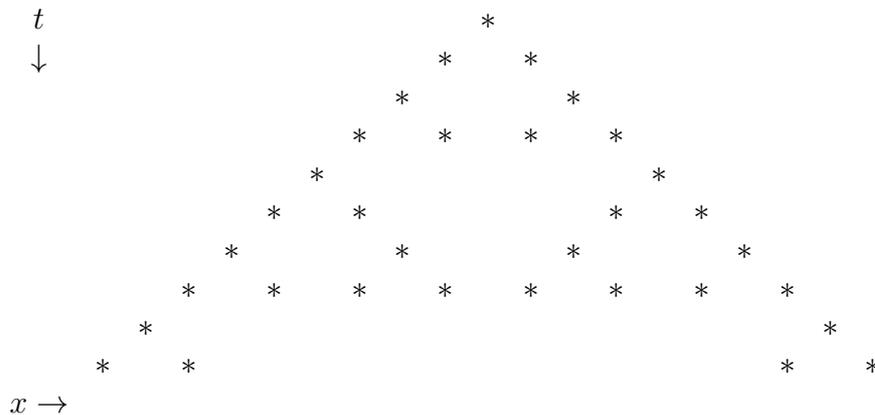


Abb. 2.2: Räumlich-zeitliches Bild eines eindimensionalen Automaten am Beispiel des SIERPIŃSKI-Fraktals

Den lokalen Zustandsüberführungen entsprechen lokale Wechselwirkungen zwischen den Zellen. Der Informationstransfer geschieht z. B. durch Diffusion oder als Wellenausbreitung und ist mit zahlreichen *Rückkopplungseffekten* verbunden. Als deren Folge kann es zur Bildung weitreichender Strukturen kommen, so beispielsweise zu Spiralwellen.

Neben den lokalen Zuständen interessieren aber auch globale quantitative Eigenschaften, etwa statistische Größen, welche sich aus den Einzelzuständen ergeben. Würde man sie in jedem Zeitschritt ermitteln, dann könnte aufgrund der damit verbundenen sequentiellen Abfrage der Zellzustände der zeitliche Gewinn, der durch der Parallelität der internen Verarbeitung entstand, wieder verloren gehen. Daher ist es ratsam, jene globalen quantitativen Eigenschaften erst am Ende der Simulation oder in größeren Zeitabständen zu bestimmen.

2.4 Ausgewählte Anwendungen

2.4.1 Biologie, Ökosysteme und LIFE

Zellulären Automaten haben nicht von ungefähr viel strukturelle Ähnlichkeit mit biologischen Organismen oder Organen, die bekanntlich selbst aus einer Vielzahl von Zellen aufgebaut sind. Jede für sich erfüllt bestimmte Funktionen (z. B. Stoffwechselprozesse oder Reizverarbeitung) und trägt so zur Gesamtfunktion des Systems bei. Auch scheint in der belebten Natur die *parallele Arbeitsteilung* in Zellen ein wesentliches Prinzip zu sein, was auch ein Grund für die Entwicklung des zellulären Automaten als ein Modell für den *Parallelcomputer* ist.

Ursprünglich versprach man sich mit den Untersuchungen zu den Selbstreproduktionsmechanismen in Zellularautomaten – die DNS wurde gerade entdeckt – sogar, mehr Verständnis über die natürlichen Vererbungsvorgänge zu gewinnen. Heute ist bekannt, daß die biologische Information als „digitales Programm“ über die Gene weitergegeben wird.

Die Ähnlichkeiten zur belebten Natur sind seit langem Motivation auch bei der Algorithmenentwicklung. Insbesondere trifft dies auf das von JOHN HORTON CONWAY im Jahre 1968 entwickelte LIFE-Spiel zu. Dieses erlangte verstärkt Popularität durch die Beiträge in der Zeitschrift „Scientific American“ in den Jahren 1970–1975. Mit diesem Spiel lassen sich grundlegende Merkmale von Lebewesen wie Geburt und Tod, Wachstum, Vermehrung und Ausbreitung simulieren. Es ist damit ein Prototyp für die Beschreibung von Populationsentwicklungen, so von Räuber-Beute-Systemen, dem Tumorstadium, der Ausbreitung von Epidemien ... Wegen der Vielfalt möglicher

Entwicklungsszenarien hatte sich LIFE zu einem unterhaltsamen Probier- und Denkspiel entwickelt, auch weil es völlig deterministisch abläuft.

Die „Spielregeln“ in LIFE sind sehr einfach: Für jede Zelle (i, j) des Raumes wird eine Umgebung zuzüglich der acht nächsten Nachbarn (MOORE-Nachbarschaft) auf einem zweidimensionalen quadratischen Gitter betrachtet. Jede Zelle nimmt zwei Zustände an. Es sei z_{ij}^t eine solche BOOLEsche Zustandsvariable mit den Werten 0 („tot“) und 1 („lebt“). Die Entwicklung der einzelnen Zellen verläuft dann einheitlich nach folgenden Regeln:

1. Ist die betrachtete Zelle zum Zeitpunkt t am Leben, dann ist sie es auch im Zeitpunkt $t + 1$ genau dann, wenn sie von zwei oder drei ebenfalls lebenden Nachbarn umgeben ist. Beträgt hingegen die Gesamtzahl lebender Nachbarn nur null oder eins, so stirbt die Zelle aufgrund von „Isolierung“. Ist andererseits diese Zahl größer als drei, so stirbt die Zelle infolge von „Überbevölkerung“.
2. Ist die betrachtete Zelle zum Zeitpunkt t hingegen tot, so wird sie im folgenden Zeitschritt $t + 1$ genau dann zum Leben erweckt, wenn sie exakt von drei lebenden Nachbarn umgeben ist.

Zusammengefaßt lauten die expliziten LIFE-Regeln für die Automatenzelle (i, j) wie folgt:

$$z_{ij}^{t+1} = \begin{cases} 1, & \text{wenn } \sum_{\text{gesamt}} z_{kl}^t = 3 \\ 1, & \text{wenn } z_{ij}^t = 1 \text{ und } \sum_{\text{gesamt}} z_{kl}^t = 4 \\ 0 & \text{sonst.} \end{cases}$$

Die Summation erstreckt sich über alle Nachbarn in der MOORE-Umgebung einschließlich der betrachteten Zelle (i, j) selbst.

Obwohl die Regeln streng determiniert sind, ist das Ergebnis der Evolution im LIFE-Raum, ausgehend von einer gewählten Startkonfiguration, meist *überraschend* und in der Regel nicht vorhersehbar. Es gibt Konfigurationen, die sich mit der Zeit stabil verhalten, und solche, die periodisch oszillieren. Man kann sowohl das Entstehen und Vergehen von „Individuen“ als auch deren flächenhafte Ausbreitung beobachten. Ein bekanntes Beispiel einer stabil sich im Raum bewegenden Konfiguration ist der sogenannte *Gleiter* (Abb. 2.3). Zudem können sich ganze „Lebensgemeinschaften“ herausbilden. Unter bestimmten Umständen beobachtet man auch Konkurrenzverhalten. Die potentielle Vielfalt an Verhaltensmustern läßt sich durch geeignete Variationen der Regeln für soziologische Fragestellungen anwenden [3].

Ein mit den LIFE-Regeln ausgestatteter zellulärer Automat besitzt *erstaunlicherweise* zugleich die Eigenschaften eines universellen Rechners. So kann er

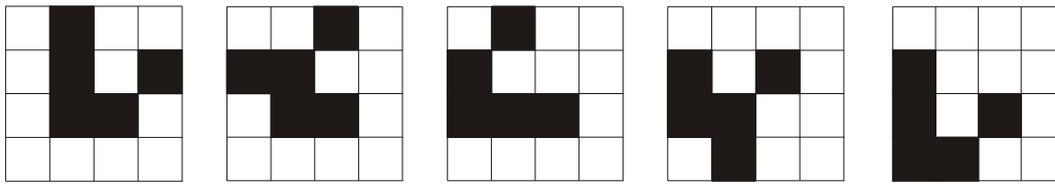


Abb. 2.3: Diagonale Fortbewegung eines „Gleiters“

Information in der Realisierung des „Gleiters“ nach den bekannten logischen Regeln verarbeiten. Als „Informationserzeuger“ dient hierbei die Konstellation der „Gleiterkanone“ (G), welche periodisch Gleiter erzeugt. Durch Kollision des einkommenden Informationsstromes mit dem Erzeugerstrom kann die Information ausgelöscht werden, wodurch die logische Operation NICHT realisierbar wird (Abb. 2.4 a). Weitere logische Operatoren leiten sich hieraus ab, so die Gatter UND sowie ODER (Abb. 2.4 b). Damit sich die nicht benötigte Infor-

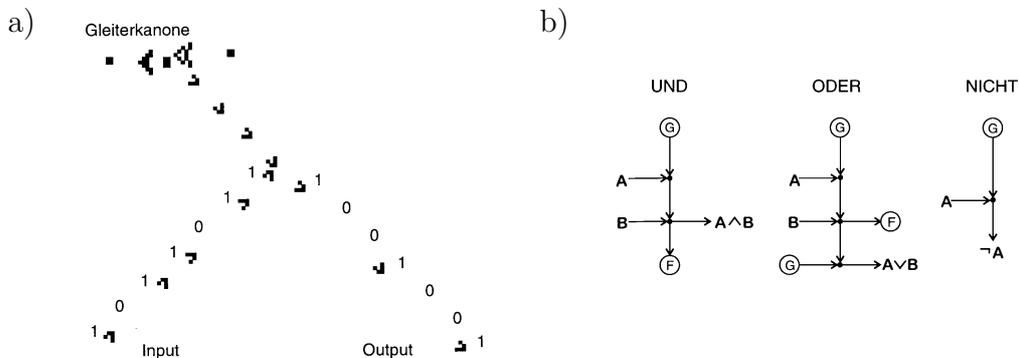


Abb. 2.4: a) „Gleiterkanone“ sowie die Negation von Information, b) die logische Operationen UND, ODER, NEGATION, wie sie mit dem LIFE-Computer realisiert werden können

mation bei weiteren Operationen nicht störend auswirkt, werden sogenannte „Freß“-Konfigurationen (F) eingesetzt. Der vollständige Aufbau eines LIFE-Computers stellt trotz der vorhandenen Grundelemente weiterhin eine architektonische Herausforderung dar. Eventuell kann zu dessen Verwirklichung eine Erweiterung auf die dritte Dimension hilfreich sein. [4]

Eine oftmals geäußerte Kritik an LIFE besteht darin, daß es bisher die Schwelle einer Datensammlung nicht überschritten hat, und eine allgemeine Theorie fehlt bislang. Ansätze zu einer statistischen Auswertung etwa der Populationszahl wurden jedoch schon in [5] vorgestellt. Als Modell für reale Ökosysteme reagiert das LIFE-System aber zu destruktiv und sensibel gegenüber

Störungen und zufälligen Einflüssen. Eine gewisse Toleranz diesen gegenüber kann man aber von einem Umweltmodell schon fordern.

Komplexere und zugleich realistischere Ökosysteme lassen sich auf ähnlich einfache Weise durch andere Automaten beschreiben. Dies wird z. B. in [6] anhand der Populationsentwicklung einer einfachen Nahrungskette Pflanze–Pflanzenfresser–Fleischfresser vorgeführt. Die Überlebenschance der Wettbewerber wird an einem einfachen Räuber-Beute-System mit jahreszeitlichem Zyklus – vereinfacht durch die Phasen Sommer und Winter – untersucht. Letzterer kontrolliert die Wachstumsrate der Pflanzen. Das Modell limitiert die Populationsentwicklung durch Min-Max-Schranken, wobei der Zuwachs bzw. das Abnehmen in diskreten Größen erfolgt. Hinzu kommt die Möglichkeit der territorialen Ausbreitung aller drei Wettbewerber.

Entsprechend den Größenlimitierungen, die als wesentliche Parameter in das Regelsystem des zellulären Automaten eingehen, können sich mit der Zeit, je nach der Stärke und Verteilung der Anfangspopulation der Pflanzen- und Fleischfresser (die Pflanzendichte wird anfänglich als konstant angesehen), in Abhängigkeit von der Dauer der Winterphase unterschiedliche Verhaltensweisen bzw. Populationsmuster herausbilden. Diese beinhalten sowohl das Aussterben beider Tierpopulationen als auch die Bildung *koexistierender* Formen. Manche Muster erinnern gar an Perserteppiche. Selbst die Entstehung *selbstorganisierender* Strukturen in Gestalt von rotierenden Spiralen oder konzentrischen Kreisen ist möglich. Sie sind das Ergebnis der Herausbildung eines ökologischen Gleichgewichts, das die Koexistenz konkurrierender Arten mit einschließt.

In [4] werden weitere Beispiele für die Populationsentwicklung, so von Haien und Fischen in einer toroidalen Wasserwelt inklusive deren Bewegung (Diffusion) als System aus Räuber und Beute, betrachtet. Auch das Problem der Bedrohung der Korallen durch Dornenkronen (Seesterne) am Great Barrier Reef, unter Berücksichtigung der hier herrschenden Strömungen und der damit verbundenen Ausbreitung der Dornenkronenlarven, wird dargelegt.

Ein anderes Modell, welches nicht nur für biologische Systeme anwendbar ist, geht auf GREENBERG und HASTINGS zurück. Es beschreibt den Wechsels von Ruhe- und Erregungszuständen mit einem dazwischenliegenden Erholungszustand etwa in Nerven- und Muskelgeweben. Aber auch die Ausbreitung von Epidemien oder von Waldbränden kann durch dieses Modell erfaßt werden. Die Entwicklung des Systems hängt wiederum stark von den Anfangsbedingungen ab. In charakteristischer Weise treten räumlich-wellenförmige Zustandsänderungen auf. Das Modell läßt sich außerdem für die Beschreibung *autokatalytischer* Reaktionsprozesse anwenden.

2.4.2 Beispiele aus der Physik

Weitere Anwendungsgebiete für zelluläre Automaten ergeben sich aus der Simulation *komplexer* räumlich-zeitlicher Prozesse, wie sie z. B. durch die lokalen Feldgleichungen der Physik beschrieben werden. Das Ziel einer Simulation mittels zellulärer Automaten besteht jedoch nicht vordergründig in der numerischen Lösung dieser Feldgleichung, sondern vielmehr in einer geeigneten Formulierung lokaler Überführungsregeln für die Automatenzelle. Hierzu werden im mikroskopischen Maßstab Zustandskonfigurationen erzeugt, die dann makroskopisch ausgewertet werden können. Die *Methodik* des zellulären Automaten ergänzt die herkömmlichen Simulationsansätze der Molekulardynamik bzw. die Monte-Carlo-Methoden. Ihr großer Vorteil besteht vor allem in der völligen Parallelität der algorithmischen Abarbeitung, da jedes diskrete Raumelement als Automatenzelle angesehen werden kann. Ein Problem, das sich aus der *Diskretisierung* des Raumes zwangsläufig ergibt, ist aber die Wahl der Größe der durch die Automatenzelle zu repräsentierenden Raumzelle.

Viele Fragestellungen, die sich mit zellulären Automaten lösen lassen, haben ihren Ursprung in der Simulation *hydrodynamischer* Probleme, so in strömungstheoretischen Anwendungsfällen mit vorgegebenen, oft komplizierten Randbedingungen wie bei Fahrzeugen oder in der Bestimmung der Ausbreitung von Schallwellen beispielsweise in Konzertsälen. Insbesondere lassen sich auch *nichtlineare* Probleme wie die Turbulenz oder die Fortpflanzung von Schockwellen untersuchen. Die Simulation von Verkehrsproblemen, z. B. der Fluß von Automobilen entlang der Autobahn oder im Stadtverkehr, zählt gleichfalls hierzu.

Transportprozesse am Beispiel der Hydrodynamik und der Diffusion

Als konkrete Beispiele sollen nun hydrodynamische Strömungsprozesse von Gasteilchen und die *kollektive* Teilchendiffusion entlang einer Geraden beschrieben werden. Aus der Verbindung dieser Transportprozesse mit chemischen Reaktionsprozessen ergäben sich weitere Anwendungsbereiche. Es sei vor allem das riesige Gebiet der *katalytischen* Prozesse genannt, das insbesondere für die technische Chemie von Interesse ist. Auch Verbrennungsvorgänge in Motoren gehören zu diesem Themenfeld.

Meist werden zur Beschreibung der Teilchendynamik bei Strömungsaufgaben sogenannte *Gittergas-Automaten* (lattice gas) betrachtet. Die Teilchen bewegen sich frei auf einem Gitter, bis sie mit anderen Teilchen zusammenstoßen. Jeder der Gitterplätze wird nur einfach besetzt. Unter der Berücksichtigung der Erhaltungssätze für die Energie und den Impuls lassen sich Regeln

für die Fortbewegung der Teilchen nach dem Stoß, die *Kollisionsregeln*, herleiten. Die zugehörigen dynamischen Gleichungen lassen sich mit Hilfe der CHAPMAN-ENSKOG-Entwicklung durch schrittweises Lösen der BOLTZMANN-Gleichung gewinnen.

Als ein erster Ansatz wurde 1973 das *HPP-Modell* (HARDY, POMEAU, DA PAZZIS, [7]) entwickelt. Die deterministischen Regeln für die molekulare Dynamik sind in der Abbildung 2.5 wiedergegeben. Die freien Teilchen bewegen sich auf diesem Gitter immer entlang der Diagonalen. Obwohl die Ergebnisse für den

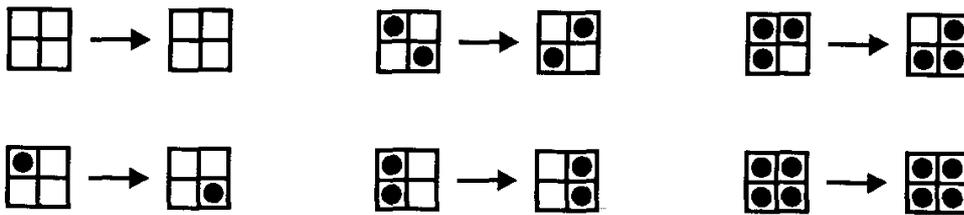


Abb. 2.5: Regeln für die molekulare Dynamik im HPP-Modell. Die übrigen Fälle ergeben sich aus Symmetrieüberlegungen. Die Regeln sind für die als MARGOLUS-Umgebung bekannte Blockstruktur definiert.

„akustischen“ Grenzfall zufriedenstellende Ergebnisse liefern, zeigt das Modell im allgemeinen hydrodynamischen Fall unphysikalische Anisotropieeffekte, die auf das Nichterfüllen bestimmter Symmetrieforderungen für die Impulse infolge des zugrunde gelegten quadratischen Gitters zurückzuführen sind.

Daraufhin wurde das *FHP-Modell* (FRISCH, HASSLACHER, POMEAU, 1968) für das hexagonale Gitter entwickelt. Dessen Stoßregeln sind in Abbildung 2.6 zu ersehen. Das Modell behebt die Defizite des HPP-Modells und gibt auch die wichtigsten physikalische Sachverhalte richtig wieder. Aber die direkte Erweiterung auf drei Raumdimensionen ist wegen des Nichtvorhandenseins einer zum hexagonalen Gitter analogen Struktur nicht möglich. Erst in vier Dimensionen existiert mit dem flächenzentrierten Hyperkubus wieder eine solche analoge Geometrie, jedoch um den Preis komplizierter Kollisionsregeln. Ergebnisse hieraus müssen dann auf die interessierenden drei Dimensionen projiziert werden.

Probleme der *Verkehrsdynamik* von Fahrzeugen auf Autobahnen und im Stadtverkehr werden gleichfalls mit hydrodynamischen Methoden untersucht, hierbei insbesondere die Abhängigkeit der Verkehrsflußdichte von der Fahrzeugdichte im sogenannten Fundamentaldiagramm. So zeigt sich auch

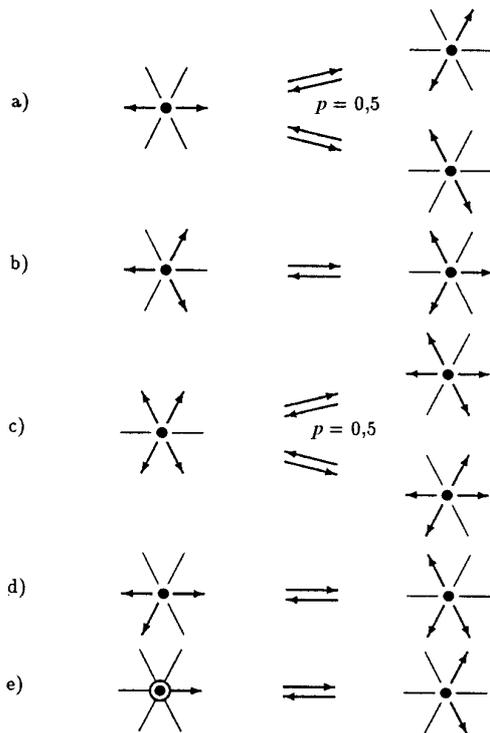


Abb. 2.6: Regeln für die molekularen Stöße im FHP-Modell

simulativ ein Maximum der Verkehrsflußdichte bei ca. 0,1 (Fahrzeuge pro Gitterplatz) mit dem ungefähren Wert von 0,32. Auch lassen sich Aussagen über das Zustandekommen von Staus und deren Vermeidung treffen. [9]

Abschließend soll ein auf der Diffusion von Teilchen begründetes Transportmodell beschrieben werden, welches auf der Vorstellung elementarer stochastischer Sprungprozesse beruht [10]. Speziell soll die kollektive Diffusion von Teilchen mit einer gegebenen Konzentration c in einem engen Kanal, praktisch entlang einer Geraden (*Single-file-Diffusion* [11]), beschrieben werden. Darin können sich die Teilchen nicht überholen. Im Unterschied zur gewöhnlichen Diffusion, bei der sich die Teilchen unabhängig bewegen, bewirkt diese geometrische Einschränkung eine zeitliche Verlangsamung der Ausbreitung. Ein Maß hierfür ist das *mittlere Verschiebungsquadrat* $\langle \vec{r}^2 \rangle$ zu asymptotisch großen Zeiten. Bei der gewöhnlichen Diffusion findet man in beliebigen Raumdimensionen eine direkte Proportionalität zur Zeit: $\langle \vec{r}^2 \rangle \sim t$. *Anomale Diffusionsprozesse*, zu denen auch die Single-file-Diffusion gehört, zeigen hingegen eine Abhängigkeit der Form $\langle \vec{r}^2 \rangle \sim t^\gamma$, wobei $0 < \gamma < 1$. Im konkreten Fall beträgt $\gamma = 1/2$,

und es gilt für große Zeiten:

$$\langle \bar{r}^2 \rangle = 2 \frac{1-c}{c} a^2 \sqrt{\frac{\Gamma t}{\pi}}. \quad (2.2)$$

Dabei ist Γ die Sprungrate des freien Einzelteilchens, a die Gitterkonstante und c die Teilchenkonzentration mit $0 < c < 1$.

Für die Simulation wurde ein *stochastisches Automatenmodell* mit sieben Zuständen entwickelt. Zur Nachbarschaft der Zelle gehören die zwei Nachbarn jeweils links und rechts. Der eigentlich unendlichen Ausdehnung des Systems wurde durch periodische Randbedingungen mit großer Periode Rechnung getragen. Die Zellenzahl beschränkt sich im konkreten Fall auf $2^{20} \approx 1$ Mio. Zellen.

Zwei getrennte Tabellen erfassen sämtliche Zustandsüberführung. Erstere beinhaltet die vier Zustände „leer“, „besetzt“, „lösche links“, „lösche rechts“ und die zweite die restlichen drei Zustände „leer“, „gehe nach links“, „gehe nach rechts“. Diese Unterteilung hat den Vorteil, daß nur noch $4^5 + 3^5 = 1267$ Tabelleneinträge gegenüber den ursprünglichen $7^5 = 16807$ für die Zustandsüberführung berücksichtigt werden müssen. Die verbleibende Vielfalt der möglichen Regeln wird durch das Verbot des Überspringens und durch die Forderung der Einfachbesetzung der Zellen weiter stark reduziert.

Die Teilchen springen jeweils nur zu den nächsten Nachbarzellen, wobei die Sprungrichtung zufällig und unabhängig von der Nachbarschaft mit Hilfe eines Zufallsgenerator bestimmt wird. Der gewählte Gitterpunkt wird besetzt, wenn er zuvor frei war und nicht vom übernächsten Nachbarn beansprucht wird. Besteht die Gefahr einer Doppelbesetzung, so entscheidet wiederum der Zufall, welcher der Konkurrenten den freien Platz einnimmt.

Im wesentlichen stimmen die Ergebnisse aus der Simulation mit dem theoretisch abgeleiteten Verhalten nach Gleichung (2.2) überein. So wurde insbesondere das Skalenverhalten der Single-file-Diffusion mit $\gamma = 1/2$ bestätigt. Geringfügige quantitative Abweichungen für das mittlere Verschiebungsquadrat sind jedoch für höhere Konzentrationen zu beobachten, die sich auf den stärker werdenden korrelativen Effekt nächster und übernächster Nachbarn im zugrunde gelegten Modell zurückführen lassen.

Mit wenig Aufwand kann das vorgestellte Modell auch auf zwei- oder dreidimensionale Gitter erweitert werden. So wurden qualitative Untersuchungen für den zweidimensionalen Fall bei konstanten Randbedingungen vorgenommen. Als Ergebnis einer solchen Simulation ist eine Momentaufnahme (Abb. 2.7) in verschiedenen Darstellungen zu sehen. Der linke und der rechte Rand wurde mit der Konzentration $c = 1$ festgelegt, am oberen und am unteren Rand

gilt hingegen $c = 0$. Aus der rechten Darstellung ist die Verwandtschaft zu einem möglichen Wärmeprofil zu erkennen, das als Folge von *Wärmeleitungs-vorgängen* entstehen würde.

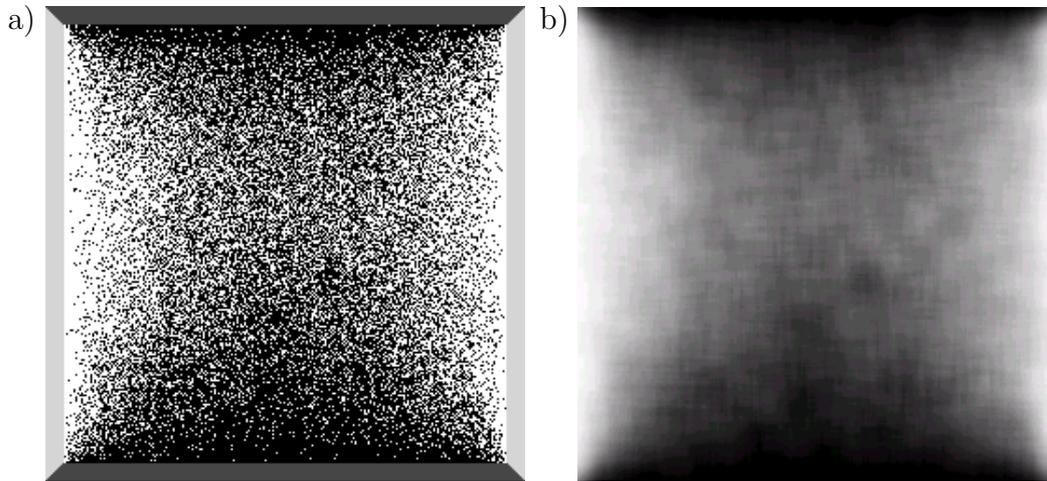


Abb. 2.7: Simulation der Teilchendiffusion mit konstanten Randbedingungen: a) in Teilchendarstellung, b) als Konzentrationsbild

Spin-Systeme

Abschließend soll die Klasse der Spin-Systeme anhand des ISING-Automaten erwähnt werden. Dieser dient zur Demonstration von Magnetisierungseffekten. Den beiden Spinzuständen des Elektrons entsprechen genau zwei Zustände in der Automatenzelle. Der physikalische Gesamtzustand (ohne äußeres Feld) wird durch die HAMILTON-Funktion H beschrieben:

$$H = -\frac{1}{2} \sum_{\substack{i,j \\ i \neq j}} J_{ij} s_i s_j. \quad (2.3)$$

Dabei ist s_i der Spinzustand am Ort i mit den Werten $+1$ oder -1 , und die J_{ij} sind die spezifischen Kopplungskonstanten der benachbarten Spins i und j . Nachfolgend wird $J_{ij} = J = \text{const.}$ angenommen. Summiert wird über alle Spins an den Orten i und den zugehörigen Nachbarn j , wobei eine VON-NEUMANN-Nachbarschaft angenommen wird.

Ausgehend von einer vorgegebenen Anfangskonfiguration, etwa einer Zufallskonfiguration, strebt das System einem thermodynamischen Gleichgewichtszustand mit minimaler Energie zu. Dem Sinnen nach völliger Ordnung – alle Spins sind zueinander parallel ausgerichtet – wirkt jedoch die thermische

Unordnung entgegen, die zum zufälliges Umklappen der Spins führen kann. Hierfür gibt es eine Übergangswahrscheinlichkeit $p(s_i \rightarrow -s_i)$, die vom lokalen Energieunterschied $\Delta H = J s_i \sum_{NN(i)} s_j$ am Ort i und der absoluten Temperatur T des Systems abhängt. Im simulierten Beispiel (Abb. 2.8) wurde die GLAUBER-Funktion

$$p(s_i \rightarrow -s_i) = \frac{1}{1 + e^{2\Delta H/k_B T}} \quad (2.4)$$

für die Zustandsüberführung gewählt (k_B : BOLTZMANN-Konstante). Unterhalb der kritischen Temperaturschwelle, der CURIE-Temperatur, beobachtet man erwartungsgemäß eine Phasenseparation, die eine bestimmte Form von *Strukturbildung* darstellt.

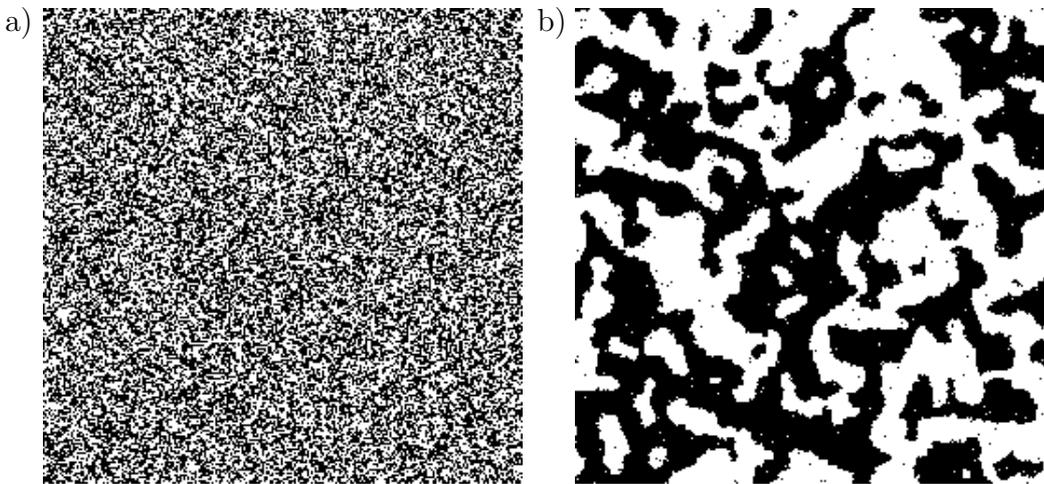


Abb. 2.8: ISING-Automat: a) eine zufällige Anfangskonfiguration, b) Phasenseparation unterhalb der CURIE-Temperatur (eine Momentaufnahme)

2.5 Hardware-Konzepte

Der Vorteil der Konzeption des zellulären Automaten liegt in der Parallelität der Abarbeitung und der damit verbundenen Erhöhung der Rechengeschwindigkeit. Die herkömmlichen sequentiell arbeitenden Rechner wurden zwar in den letzten Jahren immer schneller, der ökonomische Aufwand für die Herstellung solcher Systeme wächst aber überproportional. Sie sind daher nur als Massenware rentabel und müssen damit verbundene Anforderungen erfüllen. Eine Alternative könnten *viele kleine* parallele Logikeinheiten im Verbund einer

zellulären Struktur sein, insbesondere dann, wenn sie sich aufgabenspezifisch durch Software konfigurieren lassen.

Ein wesentlicher Aspekt, der einer *speziellen* zellulären Hardware entgegenkommt, ist der unmittelbare Datentransfer zwischen den nächsten Nachbarn. Die auszutauschenden Daten werden nicht mehr zentral gesammelt und dort weiterverarbeitet, so daß die mit langen Kommunikationswegen verbundenen technischen Probleme hier nicht auftreten können. Da die eigentliche Aufgabe zellulärer Automaten nicht in der numerischen Simulation besteht, entfällt für diese der entsprechende Hardwareaufwand.

Auch wenn sich zelluläre Automaten auf herkömmlichen Rechnern simulieren lassen und auf Parallelrechner portierbar sind, lassen sich die Vorzüge der massiven Parallelität dort entweder gar nicht oder nur mit hohem finanziellem Aufwand nutzen. Bei den sequentiell arbeitenden Rechnern kann diese Parallelität ohnehin nur simuliert werden, so daß der Rechenaufwand proportional zur Zahl der Automatenzellen ist. In Parallelrechnern hingegen werden diese auf die vorhandenen Prozessoren verteilt, und der nicht aufteilbare Anteil wird wiederum sequentiell abgearbeitet. Als zusätzliches Hemmnis taucht zudem das Problem auf, daß wegen des mit zunehmender Prozessorzahl steigenden sequentiellen Kommunikationsaufwandes die Rechenleistung nicht mehr proportional mit dieser zunimmt. Sie wächst bei vielen Prozessoren weit weniger als linear und kann im Extremfall sogar abnehmen (AMDAHLsche Gesetze). Eine der Ursachen hierfür liegt sicherlich in der notwendigen Synchronisation der Einzelprozessoren und des gegenseitigen Datenaustauschs zur Aktualisierung der Zustandsinformation.

Zelluläre Automaten können in Hinblick der *prinzipiellen* Computerarchitektur dem SIMD-Typ (SIMD: single instruction multiple data) zugeordnet werden: Die Anweisungen in Gestalt von Regeln sind für alle Zellen dieselben, und jede Zelle organisiert ihre Daten als Zustände in einem eigenen Speichersegment. Ihr einfacher Aufbau ermöglicht zahlreiche angepaßte Hardwarelösungen. Es gibt auch Ansätze, die strukturellen Vorzüge zellulärer Automaten mit denen von neuronalen Netzen zu vereinen.

Als Vorläufer zellulärer Automaten für spezielle numerische Anwendungen können die *systolischen Arrays* angesehen werden. Auch hier werden gleichartige Operationen in den Prozessorzellen durchführt, wobei aufeinanderfolgende Datenketten zyklisch durch das Operationsfeld „gepumpt“ werden. Konkrete Realisierungen zellulärer Automaten sind die Connection Machine CM-2 und die Cellular Automata Machines CAM-6 und CAM-8, wobei die CAM-6 auch kommerziell als Einsteckkarte für den PC angeboten wird (Abb. 2.9). Massivparallele Rechner wie die MASPARG oder HAL sind zwar als zelluläre Hardware

einsetzbar. Sie legen aber den Schwerpunkt auf die numerische Verwendung und sind daher ziemlich teuer.



Abb. 2.9: CAM-6 als Einsteckkarte für den PC

Es gab auch Versuche, zelluläre Automaten als rein optisches System bzw. als optoelektronische und akustooptische *Hybridlösung* zu realisieren [12], siehe Abbildung 2.10. Solche Hardwarelösungen würden vor allem wegen der schnellen optischen Datenübertragung und der Parallelität in der Ausführung Vorteile bringen. Jedoch sind die enormen Kosten und der Platzbedarf Gründe dafür, daß sich derartige Ansätze bisher praktisch nicht durchsetzen konnten.

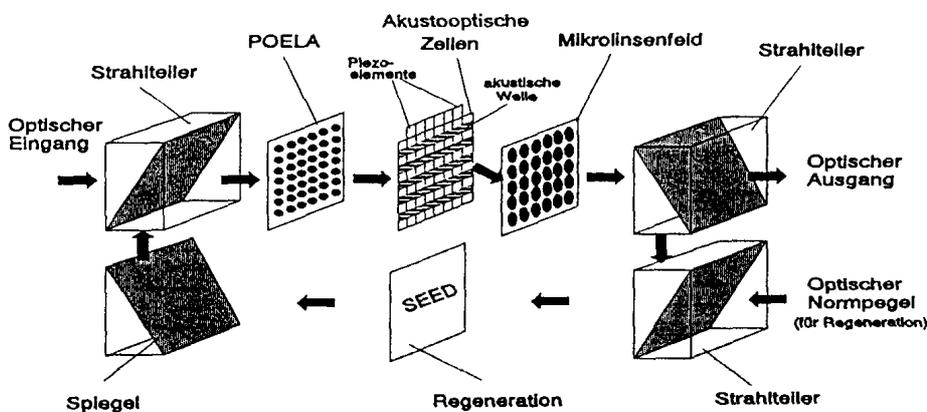


Abb. 2.10: Hybride Realisierung eines zellulären Automaten aus optischen, optoelektronischen und akustooptischen Elementen. POELA: Programmable Optical Logic Array (programmierbares optoelektronisches Logikfeld), SEED: Self Electro-Optic Effect Device (ein bistabiles optoelektronisches Element)

Weitere Ansätze verwenden zur Herstellung zellulärer Automaten programmierbare Logikbausteine, insbesondere FPGAs (FPGA: field programmable gate array). Dem Aufbau nach sind diese Bausteine hoch komplexe Speicherarrays, deren innere Architektur einem zellulären Automaten ziemlich ähnlich sieht. Mit den vorhandenen Verdrahtungsmatrizen läßt sich eine flexible Logikverschaltung erreichen, die über Software konfiguriert wird. Man erreicht so eine hohe Variabilität im Schaltungsdesign. Hardwareänderungen lassen sich einfach realisieren, und es können aufgabenspezifische Anpassungen schnell vorgenommen werden. FPGAs bilden eine mögliche Hardwarebasis für zukünftige *rekonfigurierbare Computer*, womit die bisherige Trennung von Hardware und Software zunehmend verschwindet, und zwar in dem Maße, wie es gelingt, aus einem Programm die passende Hardwarekonfiguration zu generieren. Geeignete Hardwarebeschreibungssprachen (HDL: hardware description language) gibt es hierzu bereits. Ein zugehöriger Maschinencode-Compiler müßte die softwarelogischen Operationen dann in eine passende Hardware-Sprache übersetzen.

In ferner Zukunft ist auch der Aufbau zellulärer Automaten mittels gekoppelter *Quantenpunkte* – das sind Atomcluster mit wenigen Atomen – denkbar. Als Beispiel einer solchen Automatenzelle sei eine aus fünf Quantenpunkten erzeugte Konfiguration genannt, die über Tunnelbarrieren untereinander verknüpft sind [13]. Diese wird mit zwei Elektronen beladen, so daß zwei Polarisationszustände $P = \pm 1$ herstellbar sind. Mit ihrer Hilfe lassen sich wiederum logische Gatter als grundlegende Schaltelemente herstellen, wobei deren Mehrheitslogik sich etwas von der herkömmlichen Binärlogik unterscheidet (Abb. 2.11).

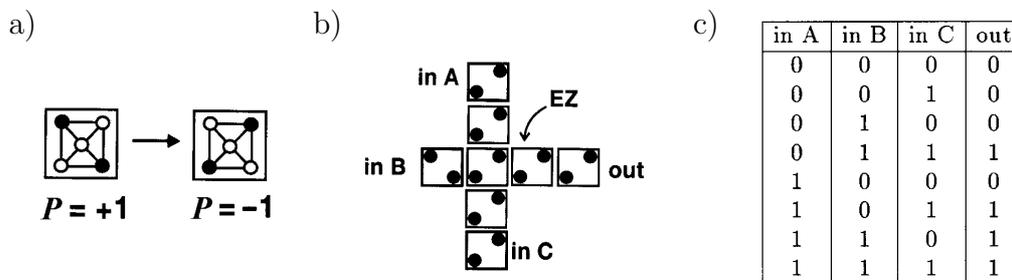


Abb. 2.11: a) Die Quantenpunkt-Zelle und b) ein aus diesen Zellen erzeugtes Gatter mit den drei Eingängen „in A“, „in B“, „in C“ und einem Ausgang „out“. c) Das logische Verhalten dieses Gatters wird in der Wahrheitstabelle vollständig erfaßt.

Literaturverzeichnis

- [1] Neuman, J. v.: *Theory of self-reproducing automata*. Edited and completed by W. Burks. Urbana: University of Illinois Press, 1966
- [2] Wolfram, S.: *Cellular Automata and Complexity*, collected papers. Reading/Massachusetts: Addison-Wesley, 1994
- [3] Klüver, J.: *Soziologie als Computereperiment*, Abschn. 2.4: *Emergierende Makrostrukturen: Zellularautomaten*. Braunschweig/Wiesbaden: Vieweg, 1995
- [4] Gerhardt, M.; Schuster, H.: *Das digitale Universum. Zelluläre Automaten als Modelle der Natur*. Braunschweig/Wiesbaden: Vieweg, 1995
- [5] Dresden, M.; Wong, D.: *Life games and statistical models*. Proc. Nat. Acad. Sci. USA 72 (1975), S. 956–960
- [6] Satoh, K.: *Computer experiment on the complex behavior of a two-dimensional cellular automaton as phenomenological model for an ecosystem*. J. Phys. Soc. Jap. 58 (1989) 10, S. 3842–3856
- [7] Hardy, J.; Pomeau, Y.; De Pazzis, O.: J. Math. Phys. 14 (1973), S. 1746
- [8] Vogeler, A.: *Hydrodynamic Simulations with Lattice Gases*. Diss. Univ. Bremen, 1992
- [9] Nagel, K.: *High speed microsimulations of traffic flow*. Inaugural-Diss., Univ. Köln, 1995
- [10] Renner, U.: *Simulation komplexer physikalischer Vorgänge durch zelluläre Automaten*. Vortrag Univ. Halle, November 1996
- [11] Kärger, J.: *Straightforward derivation of the long-time limit of the mean-square displacement in one-dimensional diffusion*. Phys. Rev. A 45 (1992) 6, S. 4173–4174
- [12] Erhard, W.; Fey, D.: *Parallele digitale optische Recheneinheiten*. Abschn. 7.4. *Exkurs: Zelluläre Automaten*. Stuttgart: B. G. Teubner, 1994
- [13] Blick, R.: *Künstliche Atome und Moleküle: Gekoppelte Quantenpunkte und Mikrowellenspektroskopie an Quantenpunkten*. Thun: Harri Deutsch, 1996