

Inhaltsverzeichnis

1	Einleitung	3
2	Definition zellularer Automaten	5
2.1	Informale Einführung	5
2.2	Formale Definition	5
2.2.1	Notwendige Begriffe	5
2.2.2	Definition	5
2.2.3	Zusammenfassung	6
2.3	Beispiele	6
2.4	Bemerkungen	7
2.4.1	Spezielle Nachbarschaften	8
2.4.2	Endliche Feldbeschränkungen	9
2.4.3	Potentiell unendliche zellulare Automaten	9
3	Wie mächtig sind zellulare Automaten?	11
3.1	Simulation von Turingmaschinen durch zellulare Automaten	11
3.2	Simulation zellularer Automaten durch Turingmaschinen	11
4	Weitere Betrachtungen	13
4.1	Umkehrbare zellulare Automaten	13
4.2	Totalistische zellulare Automaten	14
5	Ausblick	15
5.1	Random Sequence Generation	15
5.2	Kryptographie	15
5.3	Zellulare Quanten-Automaten	15
5.4	Digitale Mechanik, Digitale Physik	16
5.5	Cellular Neural Networks	16

Kapitel 1

Einleitung

Zellulare Automaten sind aus historischer Sicht das erste Modell paralleler Rechner, das eingehend untersucht wurde. Sie sind das grundlegende Modell für “massive Parallelität” und für parallele Rechner, die sehr viele Prozessoren und Array-Verbindungen zwischen Prozessoren haben.

Es gibt mächtige Zellulare-Automaten-Computer, und man erwartet, dass Zellulare-Automaten-Computer eine wichtige Rolle in der Simulation physikalischer und biologischer Prozesse spielen werden, z.B. bei der Untersuchung künstlichen Lebens. Zellulare Automaten sind heutzutage eines der wichtigen Modelle, um Phänomene der Hydro- und Gasdynamik zu untersuchen. Dafür braucht man Zellulare-Automaten-Computer, die sehr mächtig sind.

Zellulare Automaten wurden in den späten 40er Jahren von John von Neumann und Stanislaw Ulam entdeckt. Populär wurden sie jedoch erst mit der Vorstellung des “*Game Of Life*” im Jahre 1970 durch John Conway.

Während John von Neumann eher den theoretischen Fragen der zellularen Automaten nachging (“Can machines reproduce itself?”), versuchte Conway durch die spezielle Wahl der Regeln des Game Of Life das Verhalten existierender biologischer Prozesse nachzubilden. Schon 1951 zeigte von Neumann, dass es tatsächlich zellulare Automaten gibt, die selbstreproduzierend sind.

Die Standardeinführung zum Thema der zellularen Automaten ist [TofMar87], die in dieser Arbeit verwendeten Definitionen und Begrifflichkeiten lehnen sich jedoch primär an [Jant] an. Anwendungsbeispiele stammen u.a. aus [Wolf] und [GPZA]. Während des Vortrags dieser Seminararbeit wurden spezielle zellulare Automaten mit Hilfe von vorgeführten Java-Applets veranschaulicht, die in [App] und [Wolf] zu finden sind. In [CSV] sind weitere interessante Anwendungsgebiete zellulärer Automaten zu finden. Ergänzende Quellen werden im Ausblick vorgestellt.

Kapitel 2

Definition zellulärer Automaten

2.1 Informale Einführung

Informal ist ein d -dimensionaler zellulärer Automat ein d -dimensionales, unendliches Array-Netz identischer endlicher Automaten, mit einer Menge S von Zuständen und einer Übergangsfunktion δ . Alle Prozessoren (Automaten) arbeiten synchronisiert, und zu jedem Zeitpunkt $t > 0$ ist der Zustand jedes Prozessors eindeutig bestimmt durch die Übergangsfunktion und die Zustände seiner Nachbarn (bei fester Nachbarschaft N) zum Zeitpunkt $t - 1$.

2.2 Formale Definition

2.2.1 Notwendige Begriffe

Sei $(G, +)$ eine Gruppe. Sei ferner N eine Familie aus G . Für $v \in G$ definieren wir die Summe $N' = N + v$ elementweise, d.h.

$$N' = N + v \quad \text{gdw.} \quad N'_i = N_i + v \quad \text{für alle } i$$

Analog wird für eine Funktion $f : G \rightarrow G$ das Anwenden auf eine Familie N aus G definiert:

$$N' = f(N) \quad \text{gdw.} \quad N'_i = f(N_i) \quad \text{für alle } i$$

$L_d = \mathbb{Z}^d$ bezeichne die Menge aller Punkte im d -Dimensionalen Gitter. Falls d aus dem Zusammenhang klar wird, schreiben wir statt L_d auch einfach nur L .

2.2.2 Definition

Ein **zellulärer Automat** ist ein Quadrupel $A = (d, S, N, \delta)$ wobei $d \in \mathbb{N}$ die Dimension angibt, S eine endliche Menge von Zuständen ist, $N \subset L_d$ eine feste, endliche Nachbarschaft ist und $\delta : S^N \rightarrow S$ die (lokale) Übergangsfunktion.

Unter einer Konfiguration c von A verstehen wir eine Belegung aller Automaten (bzw. Gitterpunkte) mit Zuständen, also eine Abbildung $c \in S^L$.

Mit Hilfe der lokalen Übergangsfunktion δ kann die globale Übergangsfunktion $G_\delta : S^L \rightarrow S^L$ wie folgt definiert werden:

$$G_\delta(c)(v) := \delta(c(N + v)) \quad \text{wobei } c \in S^L \text{ und } v \in L$$

Eine Folge c_0, c_1, c_2, \dots aus S^L von Konfigurationen von A nennen wir **Berechnung** von A , wenn gilt:

$$c_i = G_\delta(c_{i-1}) \quad \text{für alle } i > 0$$

2.2.3 Zusammenfassung

Wir erhalten also folgende Übersicht für einen zellularen Automaten $A = (d, S, N, \delta)$:

$d \in \mathbb{N}$	Dimension
$L = \mathbb{Z}^d$	Menge aller Punkte im d -dimensionalen Gitter
S	endliche Zustandsmenge
$c \in S^L$	eine Konfiguration von A
$N \subset L$	Nachbarschaft der Zellen
$\delta : S^N \rightarrow S$	Lokale Übergangsfunktion
$G_\delta : S^L \rightarrow S^L$	Globale Übergangsfunktion

2.3 Beispiele

- Die Belegung $d = 1, N = \{-1, 0\}$ ergibt die sehr einfachen eindimensionalen *“Einweg-Zellularen-Automaten”*:



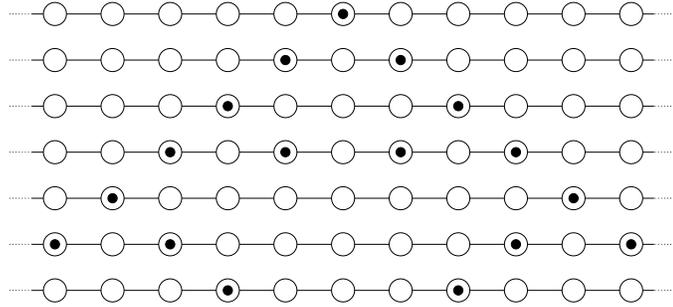
- Sei $A = (d, S, N, \delta)$ gegeben durch:

$$d = 1, \quad S = \{0, 1\}, \quad N = \{-1, 1\}, \quad \delta(x_{-1}, x_1) = x_{-1} \oplus x_1$$

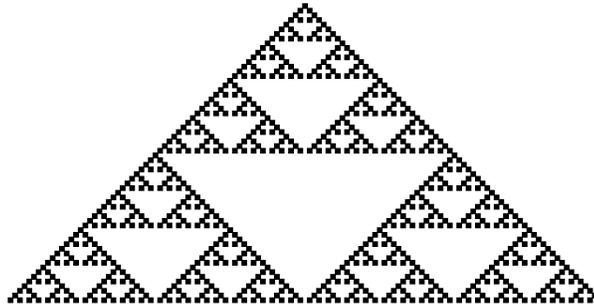
Eine Anfangskonfiguration c_0 sei gegeben durch

$$c_0(i) = \begin{cases} 1 & : i = 0 \\ 0 & : \text{sonst} \end{cases}$$

Der zeitliche Verlauf der Berechnung ab c_0 ist in der folgenden Abbildung dargestellt, wobei ein Automat im Zustand 1 mit \bullet bezeichnet wird.



Das nachfolgende Fraktal (Sierpinski-Dreieck) entsteht, wenn bei obigem Bild nur noch die Zustandswerte selbst gezeichnet werden (und der Betrachtungsausschnitt entsprechend angepasst wird). Natürlich kann der Ausschnitt nach unten beliebig weit fortgesetzt werden.



Bemerkung: Wäre für die Übergangsfunktion nicht \oplus (Addition modulo 2), sondern bei einer entsprechenden Vergrößerung der Zustandsmenge eine echte Addition (also $\delta(x_{-1}, x_1) = x_{-1} + x_1$) verwendet worden, so hätte der angegebene zellulare Automat das Pascalsche-Dreieck zeilenweise berechnet (welches also offensichtlich selbst eine fraktale Struktur aufweist!).

- Wird die Anfangskonfiguration c_0 im letzten Beispiel zufällig gewählt

$$p("c_0(i) = 0") = p("c_0(i) = 1") = 0.5 \quad \text{für alle } i$$

so entstehen über die Zeit schnell "dreieckige Muster" ("Selbstorganisation"). Diese dreieckigen Muster (siehe Abbildung 2.1) erinnern stark an das Pigmentierungsmuster einiger Schnecken (siehe Abbildung 2.2). Es ist gut möglich, dass der Wachstumsprozess dieser Pigmentierungen den Regeln eines zellularen Automaten folgt.

2.4 Bemerkungen

Zellulare Automaten sehen wie ein sehr einfaches Modell aus. Um etwas Interessantes zu berechnen, scheint es, braucht man endliche Automaten mit vielen Zuständen und einer großen Nachbarschaft. Aber schon für

$$d = 1, \quad S = \{0, 1\}, \quad N = \{-k, -k + 1, \dots, 0, \dots, k - 1, k\}$$



Abbildung 2.1: Zustands-/Zeit-Diagramm bei zufälliger Initialisierung (aus [Wolf1])

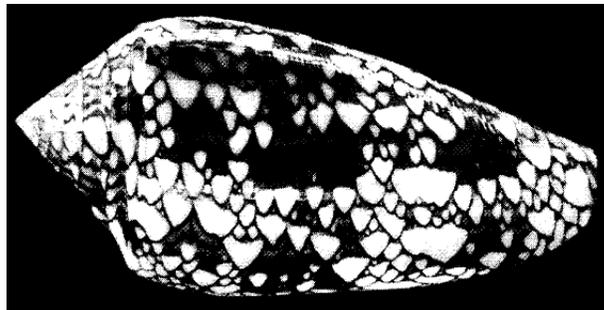


Abbildung 2.2: Pigmentierungsmuster einiger Schnecken schalen (aus [Wolf1])

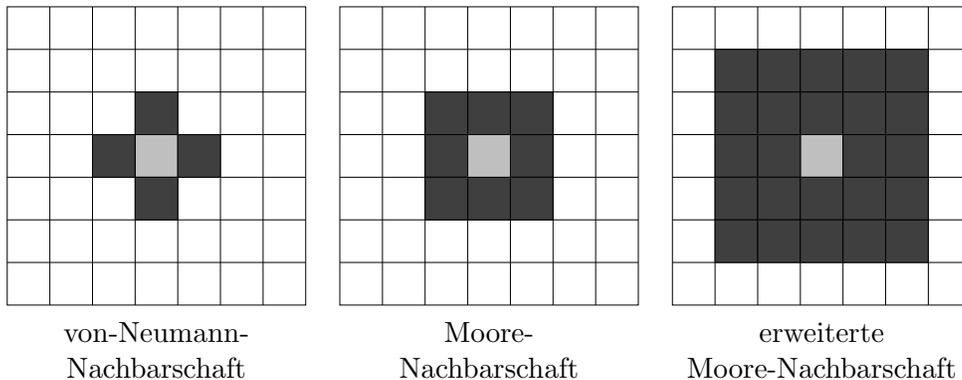
gibt es

$$2^{2^{2k+1}}$$

zellulare Automaten.

2.4.1 Spezielle Nachbarschaften

Wie in der Definition ersichtlich, kann die Nachbarschaftsrelation N eines zellularen Automaten beliebig gewählt werden. Einige spezielle haben sich bei 2-dimensionalen zellularen Automaten jedoch in der Forschung durchgesetzt.



Die **von-Neumann-Nachbarschaft** ist kreisförmig (bzgl. Vierer-Nachbarschaft) mit Radius 1. Die **Moore-Nachbarschaft** ist ebenfalls kreisförmig (bzgl. Achter-Nachbarschaft), ebenfalls mit Radius 1. Die **erweiterte Moore-Nachbarschaft** vergrößert den Radius der Moore-Nachbarschaft auf 2. Das Game Of Life verwendet die Moore-Nachbarschaft.

2.4.2 Endliche Feldbeschränkungen

Werden zellulare Automaten von Computern simuliert, tritt zwangsweise das Problem auf, daß zellulare Automaten nach Definition aus unendlich vielen Array-Automaten bestehen und aus diesem Grunde nur mit unendlich großem Speicheraufwand simulierbar sind. Um dieser Problematik auszuweichen, muß das Array beschränkt werden, also von endlicher Größe gewählt werden. Dabei treten natürlich Probleme an den Randgebieten auf, sieht die Nachbarschaftsrelation N solche ja nicht vor. Zwei Möglichkeiten bieten sich an, wobei die erste häufiger benutzt wird:

- Sich gegenüber-liegende Ränder des endlichen Arrays werden “zusammengeklebt”, d.h. miteinander identifiziert. Ein eindimensionaler zellulärer Automat wird dabei also auf einen Kreis projiziert, ein zweidimensionaler auf einen Torus.
- Ränder werden als “spiegelnd” angesehen.

Die Anzahl der Randzellen ist abhängig von der Größe des simulierten Felds: In einem 10×10 -Feld sind ca. 40% der Zellen Randzellen, in einem 100×100 -Feld nur noch um die 4%.

2.4.3 Potentiell unendliche zellulare Automaten

In vielen Anwendungen beschränkt man sich auf die Untersuchung der “potentiell unendlichen zellularen Automaten”. Bei diesen wird ein Zustand $\lambda \in S$ ausgezeichnet (genannt “sleeping state”) und sichergestellt, daß in einer Anfangskonfiguration nur endlich viele Prozessoren nicht im Zustand λ sind. Außerdem wird gefordert, daß der Nachfolgezustand einer Zelle, dessen sämtliche Nachbarn im Zustand λ sind, ebenfalls λ ist.

Eine solche Beschränkung kann z.B. bei der Simulation von Wachstumsprozessen wie bei Bakterienkulturen sinnvoll sein. Eine Petrischale wird mit einer Nährlösung gefüllt und ein “Impfbakterium” hineingesetzt. Die Bakterienkultur wächst dann von innen nach außen, nicht jedoch an Stellen, die noch unberührt sind, d.h. noch nicht von Bakterien erreicht sind. Eine unberührte Zelle wird dann mit dem Zustand λ gekennzeichnet.

Andere Beispiele wären das Wachstum einer Flechte um einen bereits vorhandenen Flechtenkern oder das Wachsen von Kristallen um einen Impfkristall.

Kapitel 3

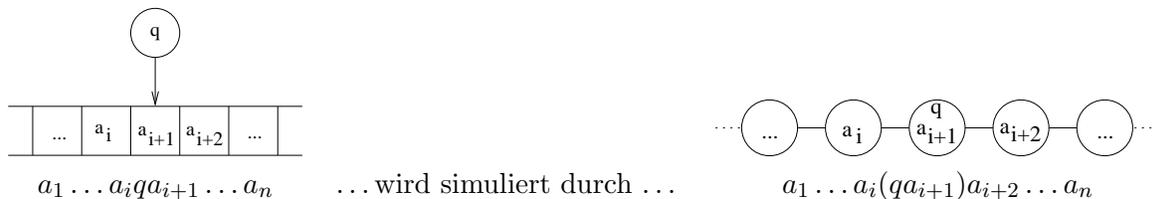
Wie mächtig sind zellulare Automaten?

Nachdem der Begriff der Rechnung eines zellularen Automaten eingeführt ist, stellen sich dem Informatiker die üblichen Fragen: wie mächtig sind zellulare Automaten, welche Klasse von Funktionen können sie berechnen? Mit welchem Aufwand (z.B. im Vergleich zur Turing-Maschine) könnten sie das tun?

3.1 Simulation von Turingmaschinen durch zellulare Automaten

Satz: Jede T -zeitbeschränkte Einband-DTM kann durch einen T -zeitbeschränkten eindimensionalen zellularen Automaten simuliert werden.

Beweis: Sei M eine Einband-DTM mit dem Bandalphabet Σ und der Menge Q von Zuständen. M wird durch einen eindimensionalen zellularen Automaten A mit der Nachbarschaft $N = \{-1, 0, 1\}$ und der Menge $\Sigma \cup (Q \times \Sigma)$ von Zuständen simuliert werden. Die M -Konfiguration $a_1 \dots a_i q a_{i+1} \dots a_n$ wird durch die Zustandsfolge $a_1 \dots a_i (q a_{i+1}) a_{i+2} \dots a_n$ kodiert. Die M -Übergangsfunktion wird durch die A -Übergangsfunktion so kodiert, dass A in einem Schritt genau einen Schritt von M simuliert (in einem Schritt ändern höchstens zwei Prozessoren ihre Zustände).

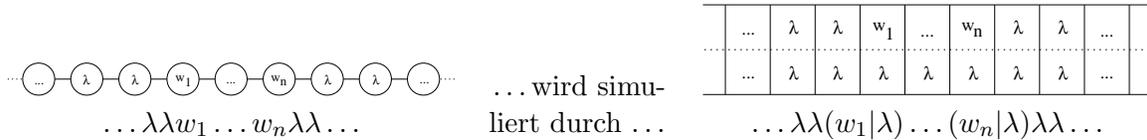


3.2 Simulation zellulärer Automaten durch Turingmaschinen

Satz: Ein eindimensionaler T -zeitbeschränkter zellulärer Automat mit der Eingabe der Länge n (d.h. alle anderen Zustände sind sleeping states) kann durch eine Einband $O((n + T(n)) \cdot T(n))$ -zeitbeschränkte DTM simuliert werden.

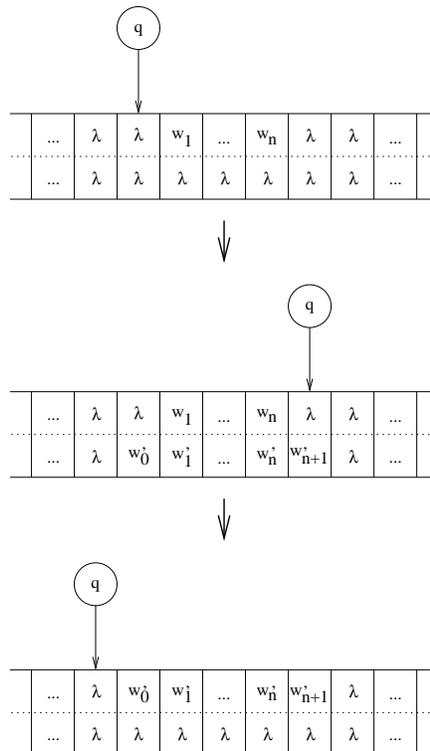
Beweis: Sei A ein eindimensionaler zellulärer Automat mit einer Nachbarschaft N und einer Anfangskonfiguration der Form $\dots \lambda \lambda w \lambda \lambda \dots$ mit $|w| = n$ und λ ist sleeping state. Wir zeigen, wie man A durch eine Einband-DTM mit zwei Spuren simulieren kann. Jeder Schritt von A wird durch eine Sequenz von Schritten von M ersetzt, in der M einmal vom linken zum rechten Ende des Bandwortes und wieder zurück läuft.

Am Anfang ist das Eingabewort auf die erste Spur geschrieben, und sein Ende ist mit den Trennsymbolen gekennzeichnet. Jede Zelle des zweispurigen Bandes ist einem Prozessor des zellulären Automaten zugeordnet. Am Anfang jeder dieser Iterationen enthält sie den aktuellen Zustand des Prozessors in der ersten Spur.



In einem Lauf von links nach rechts liest M für jeden Prozessor die Zustände seiner Nachbarn (dazu benötigt sie eine feste Anzahl Schritte) und schreibt dann den neuen Zustand auf die zweite Spur. In einem Lauf von rechts nach links kopiert M den Inhalt der zweiten Spur auf die erste Spur zurück.

Die maximale Länge der Folge der aktiven Prozessoren von A ist $O(n + T(n))$. Deshalb braucht M höchstens $O(n + T(n))$ Schritte, um einen Schritt von A zu simulieren.



Kapitel 4

Weitere Betrachtungen

Im Folgenden sollen ein paar weitere Betrachtungen zum Thema zellulare Automaten vorgestellt werden. Diese beschäftigen sich mit der Umkehrbarkeit von Rechnungen, die von zellularen Automaten gemacht wurden, und speziellen zellularen Automaten, die für Simulationen im Bereich der Physik verwendet werden.

4.1 Umkehrbare zellulare Automaten

Definition: Ein zellulärer Automat $A = (d, S, N, \delta)$ heißt umkehrbar (reversible), wenn es einen anderen zellularen Automaten $B = (d, S, N', \delta')$ gibt, so dass für jede Konfiguration c von A gilt:

$$G_\delta(c) = c_1 \quad \text{gdw.} \quad G_{\delta'}(c_1) = c$$

Anders ausgedrückt, ist ein zellulärer Automat A umkehrbar, wenn es einen anderen zellularen Automaten B gibt, so dass B für jede Berechnung von A die umgekehrte Berechnung ausführen kann.

Beispiele:

- Sei $d = 1$, $S = \{0, 1\}$. Wenn $N = 0$ und $\delta : S \rightarrow S$ eine Bijektion ist, dann ist $A = (d, S, N, \delta)$ ein (trivialer) umkehrbarer Automat.
- Wenn $|N| = 2$ oder $|N| = 3$ gilt, so gibt es keinen umkehrbaren zellularen Automaten mit der Nachbarschaft N .
- Von den 2^{16} zellularen Automaten mit einer Nachbarschaft $|N| = 4$ sind nur acht umkehrbar.

Satz: Es gibt universelle umkehrbare zellulare Automaten (d.h. sie können jede Turingmaschine simulieren).

Satz: Es ist entscheidbar (unentscheidbar), ob ein eindimensionaler (zweidimensionaler) zellulärer Automat umkehrbar ist.

4.2 Totalistische zellulare Automaten

Viele zellulare Automaten, die wichtig für die Physik sind, sind “totalistisch”.

Definition: Ein zellulärer Automat $A = (1, S, \{-1, 0, 1\}, \delta)$ heißt totalistisch, wenn $S \subset \mathbb{N}$, $|S| < \infty$ ist und es eine Funktion $\Phi : \mathbb{N} \rightarrow \mathbb{N}$ gibt, so dass gilt:

$$\delta(x, y, z) = \Phi(x + y + z)$$

Definition: Wir schreiben $\boxed{x = \text{“}a_n a_{n-1} \dots a_0 \text{“}_b}$ falls $b \in \mathbb{N}$ und es Zahlen a_0, a_1, \dots, a_n gibt mit $a_k \in \{0 \dots b-1\}$ und $x = \sum_{k=0}^n b^k a_k$

Satz: Für jeden zellulären Automaten $A = (1, S, \{-1, 0, 1\}, \delta)$ gibt es einen zellulären Automaten $B = (1, \bar{S}, \{-1, 0, 1\}, \bar{\delta})$, der totalistisch ist und A ohne Zeitverlust simulieren kann.

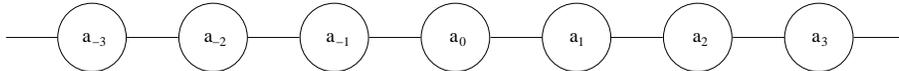
Beweis: Sei $b = |S| + 1$ und o.B.d.A. gelte $S = \{1, 2, \dots, b-1\}$. Dann wird \bar{S} festgelegt durch

$$\bar{S} = \left\{ b^k \cdot s \mid s \in S \text{ und } k \in \{0, \dots, 3\} \right\}$$

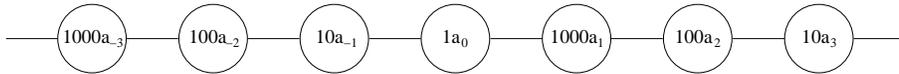
oder in b -närer Schreibweise:

$$\bar{S} = \{m \cdot s \mid s \in S \text{ und } m \in \{1_b, 10_b, 100_b, 1000_b\}\}$$

Eine A -Konfiguration



wird dann simuliert durch



Mit der Abkürzung $\bar{x} = x_{-1} + x_0 + x_{+1}$ wird nun $\bar{\delta}$ durch Φ definiert als

$$\bar{\delta}(x_{-1}, x_0, x_{+1}) = \Phi(\bar{x}) = \begin{cases} 10 \cdot \delta(k, l, m) & \text{falls } \bar{x} = \text{“}0klm\text{“}_b \\ \delta(l, m, k) & \text{falls } \bar{x} = \text{“}k0lm\text{“}_b \\ 1000 \cdot \delta(m, k, l) & \text{falls } \bar{x} = \text{“}kl0m\text{“}_b \\ 100 \cdot \delta(k, l, m) & \text{falls } \bar{x} = \text{“}klm0\text{“}_b \end{cases}$$

Kapitel 5

Ausblick

Neben den vorgestellten klassischen zellularen Automaten wurden in der Geschichte auch Variationen, sowohl der Automaten selbst, als auch der Einsatzgebiete, untersucht. Einige sollen hier kurz vorgestellt werden:

5.1 Random Sequence Generation

Stephen Wolfram, einer der zentralen Erforscher der zellularen Automaten, suchte nach Möglichkeiten, wie sich zellulare Automaten zur Erzeugung (pseudo-)zufälliger Muster nutzen ließen. Eine zentrale Idee beruht dabei auf den eindimensionalen zellularen Automaten mit zufälliger Initialisierung (“Initialisierungsvektor”) und nicht-linearer Übergangsfunktion δ . Wolfram fand Automaten, die zufällige Sequenzen mit recht guten statistischen Eigenschaften erzeugen konnten. Nähere Informationen unter

<http://www.stephenwolfram.com/publications/articles/ca/86-random/>

5.2 Kryptographie

Generatoren (pseudo-)zufälliger Folgen lassen sich unter Umständen zur Konstruktion von Strom-Chiffren verwenden. Dabei wird ein zufälliger Bitstrom erzeugt (in Abhängigkeit von einem Initialisierungsvektor) und dieser bitweise mit dem Quelltext verXORt. Nur ein Empfänger, der den (geheimen) Initialisierungsvektor kennt, kann den selben Bitstrom erzeugen, und das Chiffre wieder sichtbar machen. Wolfram untersucht die Verwendungsmöglichkeiten von zellularen Automaten für die Kryptographie in

<http://www.stephenwolfram.com/publications/articles/ca/85-cryptography/>

5.3 Zellulare Quanten-Automaten

Die theoretische Informatik modelliert Quantencomputer durch endliche Automaten, deren Kanten komplexwertige Gewichte haben. Die Automaten schalten nicht deterministisch, sondern mit Wahrscheinlichkeiten, die von den jeweiligen Gewichtungen bestimmt sind. Es ist möglich das Modell zellulärer Automaten entsprechend zu erweitern: die zellularen Quanten-Automaten sind Arrays von Quanten-Automaten. Sie werden genauer untersucht in

<http://www.cs.berkeley.edu/~vandam/qca.ps.gz>

5.4 Digitale Mechanik, Digitale Physik

Zellulare Automaten benutzt man heutzutage um verschiedene physikalische Phänomene zu modellieren, die gewöhnlich durch partielle Differentialgleichungen dargestellt werden.

Differentialgleichungen formen die mathematische Grundlage für die Mehrzahl der Modelle natürlicher Systeme. Die zellularen Automaten kann man betrachten wie eine alternative – und in gewisser Weise ergänzende – Basis für das mathematische Modell.

Für das Studium des Modells zellulärer Automaten von der physikalischen Welt wurde der Ausdruck “digitale Mechanik” geprägt. E. Fredkin, der Begründer der digitalen Mechanik, postuliert, dass ein einzelner zellulärer Automat gefunden werden wird, der die gesamte mikroskopische Physik exakt modelliert.

Ein gewisser Fortschritt ist heutzutage schon erreicht, indem gezeigt wurde, wie verschiedene Aspekte der Physik mit zellularen Automaten modelliert werden können.

Da die mikroskopische Physik umkehrbar ist, konzentriert man sich auf die umkehrbaren (universellen) zellularen Automaten, in denen die Information und andere Quantitäten in natürlicher Weise erhalten bleiben.

http://digitalphilosophy.org/dm_paper.htm

<http://digitalphysics.org>

5.5 Cellular Neural Networks

Ähnlichkeiten und Unterschiede zwischen zellularen Automaten und neuronalen Netzen, sowie deren Kombinationsmöglichkeiten, werden untersucht in

www.ce.unipr.it/pardis/CNN/cnn.html

Literaturverzeichnis

- [TofMar87] Tommaso Toffoli, Norman Margolus: **Cellular automata machines: a new environment for modeling**, ISBN: 0-262-20060-0
- [Toff76] Tommaso Toffoli: **Computation- and construction-universality of reversible cellular automata**, ReportNr: TR LCG76 192
- [Neum66] John von Neumann: **Theory of self-reproducing automata**, Univ. of Ill. Press
- [JCSS75/2] **Journal of Computer & System Sciences**, Vol 11, Number 2, Oct 1975, ISSN: 0022-0000
- [JCSS75/3] **Journal of Computer & System Sciences**, Vol 11, Number 3, Dec 1975, ISSN: 0022-0000
- [Jant] Matthias Jantzen: **Theoretische Grundlagen der Programmierung**, Vorlesungs-Skript, Universität Hamburg
- [App] **Allgemeine Infos zu zellularen Automaten, Applets**
<http://cgi.student.nada.kth.se/cgi-bin/d95-aeh/get/lifeeng>
- [CSVL] **Complex Systems Virtual Library**
http://life.csu.edu.au/vl_complex/0CellularAutomata.html
- [Wolf] **Stephen Wolfram's Publications**
<http://www.stephenwolfram.com/publications/articles/ca/>
- [Wolf1] **The Nature of Cellular Automata and a Simple Example**
<http://www.stephenwolfram.com/publications/articles/ca/83-cellular/2/text.html>
- [GPZA] **Geschichtliches und Philosophisches zu zellularen Automaten**
<http://www.ifs.tuwien.ac.at/~aschatt/info/ca/ca.html>