

Martin Sollert
Rubensstraße 6
49716 Meppen
Tel.:05931-2288
EMail:martin.sollert@ewetel.net

Meppen, im Dezember 2006

Algorithmische Klassifikation eindimensionaler zellulärer Automaten mit symmetrischem Regelsatz

Inhaltsverzeichnis:

- 0.Einleitung
- 1.Grundlagen
- 2.Der Klassifikationsalgorithmus
- 3.Die Beurteilung des Klassifikationsalgorithmus
- 4.Anwendungen des Klassifikationsalgorithmus
- 5.Schlussbetrachtung
- 6.Anhang
- 7.Literaturverzeichnis

0.Einleitung

Nach Stephen Wolfram teilt man eindimensionale zelluläre Automaten in vier Klassen ein (siehe z.B. [Wo1], S. 115-157 oder [Wo2], S.231-296 oder [GeSch], S.72-92). Ein zellulärer Automat (im Folgenden auch häufig mit ZA abgekürzt) der Klasse I hat die Eigenschaft, dass alle Zellen nach nur wenigen Zeitschritten einen einheitlichen Zustand annehmen. ZA der Klasse II erzeugen bei zufälliger Startbelegung nach wenigen Zeitschritten stationäre Muster oder sich zyklisch in relativ kurzen Perioden wiederholende relativ einfache Muster. Klasse III - Automaten bilden bei ihrer Entfaltung nichtstationäre Muster aus, die chaotisch anmuten und in der Regel relativ fein strukturiert sind. Ihr Zeitverhalten erscheint chaotisch und nicht prognostizierbar zu sein. Die Automaten der Klasse IV sind am interessantesten und auch selten. Ihr Raum-Zeit-Verhalten lässt sich am ehesten mit dem Adjektiv "komplex" umschreiben. Sie weisen einerseits deutliche Ordnungsstrukturen, andererseits aber auch chaotisch und regellos anmutende Raum-Zeit-Muster auf. In einigen Raumbereichen ist manches Mal gute Vorhersagbarkeit gegeben, in anderen Raumbereichen dagegen kaum oder gar nicht, und die betreffenden Raumbereiche ändern ihre Lage im Zellraum i.a. mit fortschreitender Zeit. Die erzeugten Raum-Zeit-Muster liegen in einer Art Zwischenbereich zwischen ZA der Klassen II und solchen der Klasse III.

Nicht alle Automaten lassen sich eindeutig einer der vier Klassen zuordnen. Auf dieses Problem hat schon Stephen Wolfram selbst hingewiesen (siehe [Wo2], S.240). In den allermeisten Fällen gibt es jedoch keine Zweifel über die Zuordnung. Die Einordnung eines gegebenen ZA in eine der vier Wolframschen Klassen trifft ein menschlicher Beobachter. Dazu beobachtet er die Raum-Zeit-Entfaltung des Automaten und nimmt daran die Einstufung vor. Ich nenne dies die subjektive Klassifikation. Die subjektive Klassifikation erfolgt intuitiv. Erstaunlicherweise decken sich die Einschätzungen verschiedener Beobachter in der überwiegenden Zahl aller Fälle.

Interessant für viele Untersuchungen zu zellulären Automaten sind vor allem die komplexen Automaten der Klasse IV. Um sie zu finden, muss ein menschlicher Beobachter systematisch oder auch zufällig Regelsätze durchprobieren, um durch subjektive Klassifikation solche Automaten aufzufinden. Diese Durchmusterung kann sehr arbeitsaufwändig sein, da die Automaten der Klasse IV bei relativ großen Zellzustandsanzahlen extrem selten sind. Somit taucht die Frage auf, ob es möglich ist, die Einordnung eines ZA in eine der vier

Wolframschen Klassen zu automatisieren, also einen Algorithmus zu konstruieren, der unabhängig von einem menschlichen Beobachter die Klassifikation des ZA vornimmt. Ich nenne dies die algorithmische Klassifikation. Andy Wuensche hat 1998 solch einen Algorithmus vorgeschlagen (siehe [Wu]). Dieser Algorithmus beruht auf Berechnungen der zeitlichen Schwankung der Entropie der Häufigkeitsverteilung der einzelnen Bestandteile des jeweils dem ZA zugrundeliegenden Regelsatzes. Meines Wissens nach ist dieser Algorithmus jedoch nicht zu einer systematischen Durchmusterung zwecks algorithmischer Klassifikation eingesetzt worden, um beispielsweise zu Aussagen über Häufigkeitsverteilungen über die vier Klassen innerhalb einer gegebenen Nachbarschafts-Zustandsanzahl-Konfiguration zu gelangen.

In dieser Arbeit wird ein alternativer Ansatz zur Entwicklung eines Klassifikationsalgorithmus verfolgt, der hauptsächlich für solche eindimensionale ZA (im Folgenden auch als eZA bezeichnet) funktioniert, deren Regelsatz symmetrisch bezüglich der Nachbarschaftsstruktur ist (d.h. $f[\text{Links},\text{Mitte},\text{Rechts}] = f[\text{Rechts},\text{Mitte},\text{Links}]$, wenn f die den Regelsatz beschreibende Funktion des ZA ist). Bei asymmetrischem Regelsatz arbeitet der Klassifikationsalgorithmus auch, jedoch mit einer geringeren Sicherheit. Er beruht im wesentlichen auf einer speziellen Art der Datenkompression bezüglich der Zellzustände eines eindimensionalen ZA. Der Klassifikationsalgorithmus wird zunächst detailliert entwickelt und anschließend hinsichtlich seiner Güte untersucht, d.h. es wird die Frage beantwortet, inwieweit subjektive und algorithmische Klassifikation bei eindimensionalen ZA mit symmetrischem Regelsatz übereinstimmen (solche ZA sollen im Folgenden auch esZA genannt werden).

Danach wird der Klassifikationsalgorithmus selbst als Werkzeug zu Untersuchungen hinsichtlich esZA eingesetzt:

1. Mit Hilfe des Klassifikationsalgorithmus wird die Suche nach Klasse IV - Automaten automatisiert.
2. Es wird die Verteilung der vier Wolframschen Klassen im Regelsatzraum eines gegebenen Automatentyps (alle esZA mit gleicher Nachbarschafts- und Zellzustandsanzahl sind vom gleichen Automatentyp; die zugehörigen Automatenregelsätze bilden den sogenannten Regelsatzraum) unter Anwendung des Klassifikationsalgorithmus ermittelt. Es zeigt sich, dass diese Verteilung signifikant vom Automatentyp abhängt und insbesondere Entwicklungen hinsichtlich der Zellzustandsanzahl aufweist.
3. Es wird mit Hilfe des Klassifikationsalgorithmus untersucht, inwieweit Klasse IV - Automaten Häufungen im Regelsatzraum eines bestimmten Automatentyps bilden. Man stellt fest, dass mit wachsender Zellzustandsanzahl eine zunehmende Tendenz zur „Klumpigkeit“ besteht.

1.Grundlagen

Ein eindimensionaler zellulärer Automat besteht aus sogenannten Zellen, die nebeneinander angeordnet sind. Theoretisch kann der Automat unendlich viele Zellen besitzen. Aus praktischen Gründen betrachtet man jedoch immer nur Automaten mit endlicher Zellenanzahl. Die Zellenanzahl soll auch Weltbreite wb des Automaten genannt werden. Die Zellen werden der Reihe nach von 0 bis zur Zahl $wb-1$ durchnummeriert:

$$c_0 , c_1 , c_2 , \dots , c_{wb-2} , c_{wb-1}$$

Jede Zelle kann endlich viele Zustände $0,1,2,\dots,z-1$ annehmen, wobei z eine natürliche Zahl größer als 1 ist und als Zellzustandsanzahl bezeichnet wird. Die Menge $Z = \{0,1,2,\dots,z-1\}$ soll Zustandsmenge des Automaten heißen. Die einzelnen Zellen sind nebeneinander angeordnet, wobei jede Zelle die gleiche Anzahl von sogenannten Nachbarzellen auf jeder Seite besitzen soll. Mit der Variablen n soll die Anzahl der Nachbarzellen auf einer Seite einer Zelle bezeichnet werden. Eine Zelle soll definitionsgemäß auf jeder Seite die gleiche Anzahl von Nachbarzellen haben, d.h. die Nachbarschaftsstruktur soll bezüglich einer Zelle stets symmetrisch sein. Eine Zelle selbst gehört immer zu ihrer eigenen Nachbarschaft. Somit besitzt eine Zelle immer insgesamt $2 \cdot n + 1$ Nachbarzellen. Die folgende Grafik Abb.1.1 verdeutlicht dies an einem Beispiel:

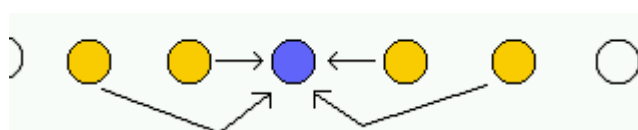


Abb.1.1

Die betrachtete blau eingefärbte Zelle besitzt auf jeder Seite zwei Nachbarzellen ($n = 2$), so dass insgesamt 5 Zellen ihre Nachbarschaft bilden. Es werden im Folgenden nur solche ZA betrachtet, bei denen alle Zellen

des Automaten die gleiche Nachbarschaftsstruktur aufweisen.

Die Endlichkeit der Weltbreite erfordert eine sinnvolle Konvention bezüglich der Ränder eines ZA, also der Festlegung der Nachbarzellen derjenigen Zellen mit kleinen Nummern bzw. Nummern nahe bei $wb-1$. Es soll, wie in der Literatur üblich, der ZA an den Rändern gewissermaßen 'zusammengeklebt' werden, damit er eine zyklische Anordnung erhält:

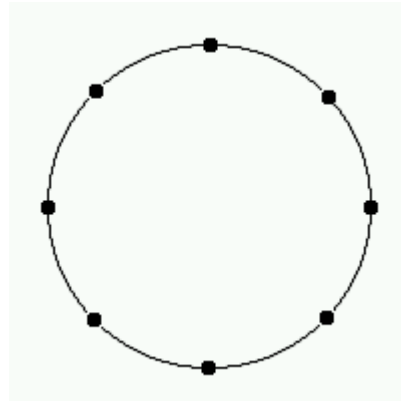


Abb.1.2

Somit ist gewährleistet, dass jede Zelle des endlichen ZA die gleiche Nachbarschaftsstruktur besitzt. Im Beispiel aus Abb.1.1 bedeutet dies für den Fall $wb=100$:

Die Zelle c_0 hat rechts die Nachbarzellen c_1 und c_2 und links die Nachbarzellen c_{99} und c_{98} .

Die Zelle c_1 hat rechts die Nachbarzellen c_2 und c_3 und links die Nachbarzellen c_0 und c_{99} .

Die Zelle c_{99} hat rechts die Nachbarzellen c_0 und c_1 und links die Nachbarzellen c_{98} und c_{97} .

Die Zelle c_{98} hat rechts die Nachbarzellen c_{99} und c_0 und links die Nachbarzellen c_{97} und c_{96} .

Ihre Dynamik gewinnen ZA dadurch, dass sich die Zellzustände ihrer Zellen mit fortschreitender Zeit gemäß exakt festgelegter Regeln entwickeln. Diese Regeln werden im sogenannten Regelsatz des Automaten zusammengefasst. Alle Zellen des Automaten besitzen den gleichen Regelsatz. Befindet sich der Automat zum Zeitpunkt t in der Zellzustandskonfiguration

$$C(t) = [c_0(t), c_1(t), c_2(t), \dots, c_{wb-2}(t), c_{wb-1}(t)],$$

so wird der Regelsatz auf jede der wb Zellen angewandt. Eine Zellgeneration später, also zum Zeitpunkt $t+1$, befindet sich der ZA in der neuen Zellzustandskonfiguration

$$C(t+1) = [c_0(t+1), c_1(t+1), c_2(t+1), \dots, c_{wb-2}(t+1), c_{wb-1}(t+1)]$$

Der Regelsatz ist eine Funktion $f: Z^{2 \cdot n+1} \longrightarrow Z$.

Jedem $(2 \cdot n+1)$ -Tupel einer möglichen Nachbarschaftszustandskonfiguration wird eine Zahl aus Z zugeordnet, die den Zellzustand der Zelle in der nächsten Generation beschreibt, falls in der vorhergehenden Generation die Nachbarschaftszustandskonfiguration vorliegt, die dem jeweiligen $(2 \cdot n+1)$ -Tupel entspricht. Der Zustand einer Zelle in der nächsten Generation ist damit also stets eine Funktion der Zustände ihrer Nachbarzellen in der jetzigen Generation. Der ZA besteht damit einerseits aus den Zellen und andererseits aus dem Regelsatz, der die zeitliche Veränderung der Zellzustände seiner Zellen beschreibt.

Ein Beispiel soll das Gesagte verdeutlichen:

Es gelte $Z = \{0, 1\}$ (also $z = 2$), $n = 1$ und $wb = 8$.

Der Regelsatz werde wie folgt definiert:

$$\begin{aligned} f([0,0,0]) &= 0; & f([0,0,1]) &= 1; & f([0,1,0]) &= 1; & f([0,1,1]) &= 1; & f([1,0,0]) &= 0; \\ f([1,0,1]) &= 1; & f([1,1,0]) &= 1; & f([1,1,1]) &= 0 \end{aligned}$$

Der ZA soll die anfängliche Zellzustandskonfiguration

$$C(0) = [1, 0, 0, 1, 0, 0, 0, 1]$$

besitzen.

Die Zelle c_0 befindet sich im Zustand 1. Die rechte Nachbarzelle von Zelle c_0 , nämlich Zelle c_1 , befindet sich im Zustand 0. Die linke Nachbarzelle von Zelle c_0 , nämlich Zelle c_7 , befindet sich im Zustand 1. Also ist die Zelle c_0 in der nächsten Generation gemäß des obigen Regelsatzes im Zustand $f([1,1,0])=1$. Entsprechend gelangt man zu den Zellzuständen der übrigen Zellen in der nächsten Generation. Es ist also

$$C(1) = [1, 0, 1, 1, 0, 0, 1, 1] .$$

Analog folgt

$$C(2) = [1, 1, 1, 1, 0, 1, 1, 0]$$

und

$$C(3) = [1, 0, 0, 1, 1, 1, 1, 1]$$

und so fort.

In der Regel haben die betrachteten ZA eine Weltbreite in der Größenordnung von mehreren Hundert und die möglichen Zustände werden zur Visualisierung durch verschiedene Farben dargestellt. In dieser Arbeit werden die folgenden Farbzuordnungen verwendet:

Der Zustand 0 wird stets mit der Farbe weiss kodiert und der höchste Zustand $z-1$ stets mit der Farbe schwarz. Falls mehr als zwei Zustände vorliegen, werden die übrigen Zustandswerte aufsteigend mit den vom .NET Framework von Microsoft computergenerierten RGB-Farben Yellow, Red, LightGreen, Blue, Brown, Cyan, LightGray, Aquamarine, Chocolate, DarkCyan, DarkGreen, Pink, Gold, LightYellow, LightSalmon, SandyBrown, Thistle, Plum, usw. kodiert.

Wird der oben definierte Regelsatz auf einen Zellraum der Weltbreite 200 insgesamt 200 Mal iterativ angewendet, so ergibt sich bei zufälliger Startbelegung der 200 Zellen (d.h. die ursprüngliche Zellzustandskonfiguration zum Zeitpunkt $t=0$ wird zufällig gewählt) beispielsweise das folgende Bild:

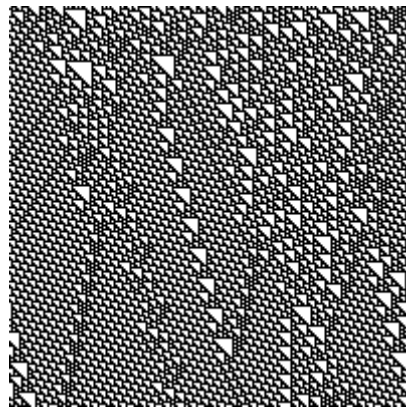


Abb.1.3

Bei einer anderen zufälligen Startbelegung ergibt sich beim gleichen Automaten für die ersten 200 Zellgenerationen das folgende Bild:

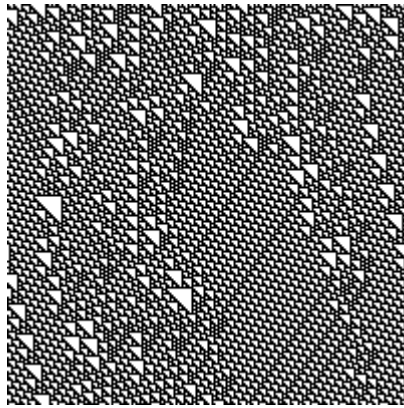


Abb.1.4

Jede der 200 Linien beschreibt jeweils die Zellzustandskonfiguration $C(t)$ zu einem gegebenen Zeitpunkt t , also die räumliche Verteilung der Zellzustände über den Zellraum. Durch das Aneinanderlegen der Zellzustandskonfigurationen zu aufeinanderfolgenden Zeitpunkten wird die zeitliche Entwicklung der räumlichen Verteilung der Zellzustände über den Zellraum sichtbar. Ändert man den Regelsatz des ZA ab, so entsteht ein (u.U. völlig) abweichendes räumlich-zeitliches Muster. So zeigt die nachfolgende Grafik Abb.1.5 das entsprechende Muster, wenn man beim obigen Regelsatz lediglich den Regelbestandteil $f[(0,1,0)]=1$ gegen $f[(0,1,0)]=0$ austauscht und alle übrigen 7 Regelbestandteile beibehält:

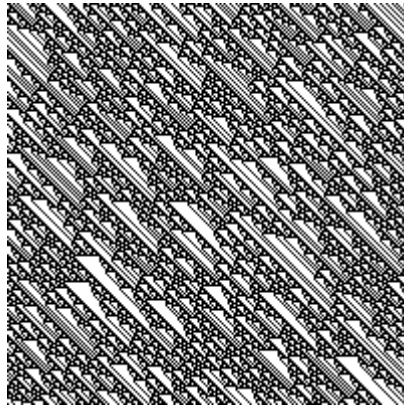


Abb.1.5

Es ist ein völlig anderes Muster als das aus den Abbildungen 1.3 und 1.4 entstanden. Wählt man andere zufällige Startbelegungen für diesen ZA, so sieht das Muster dem in Abb.1.5 sehr ähnlich. Es weist die gleichen Strukturmerkmale auf. Jeder Automat erzeugt also für ihn spezifische Raum-Zeit-Muster, die ihn gewissermaßen kennzeichnen. Jeder Automat entfaltet sich auf spezifische Weise, die durch seinen Regelsatz (und natürlich auch durch seine Startbelegung) festgelegt wird. Ich spreche deshalb im Folgenden von der Raum-Zeit-Entfaltung des Automaten.

Bei vorgegebener Zustandsanzahl z und Nachbarschaftsanzahl $2 \cdot n + 1$ gibt es i.a. eine große Fülle möglicher Regelsätze für einen ZA. Um einen ZA mit seinem zugehörigen Regelsatz eindeutig und effektiv zu kennzeichnen, ist eine kurze und eindeutige Notation erforderlich. Diese wird nun entwickelt. Sie entspricht weitgehend der in der Fachliteratur üblichen Notation.

Zunächst wird durch „eZA“ ausgedrückt, dass es sich um einen eindimensionalen ZA handelt. Danach erfolgt die Angabe von n und z in der Schreibweise „ $nNzZ$ “. Automaten mit der Notation eZA_{3N5Z} sind also eindimensionale zelluläre Automaten mit $n=3$ und $z=5$. Alle eZA mit gleichem n und z bilden den sogenannten Automatentyp. Für jeden Automatentyp existiert die Menge der Regelsätze, die natürlich von n und z abhängt. Erst durch die zusätzliche Angabe des Regelsatzes wird der betreffende Automat eindeutig spezifiziert.

Betrachtet man im obigen Regelsatzbeispiel die möglichen Nachbarschaftszustandskonfigurationen, die als acht 3-Tupel notiert sind, also

$$f([0,0,0])=0; f([0,0,1])=1; f([0,1,0])=1; f([0,1,1])=1; f([1,0,0])=0; \\ f([1,0,1])=1; f([1,1,0])=1; f([1,1,1])=0$$

so lassen sich die acht 3-Tupel als binär codierte Zahlen auffassen, die der Größe nach angeordnet sind:

$$\begin{aligned} [0,0,0] &\leftrightarrow 000_2 \\ [0,0,1] &\leftrightarrow 001_2 \\ [0,1,0] &\leftrightarrow 010_2 \\ [0,1,1] &\leftrightarrow 011_2 \\ [1,0,0] &\leftrightarrow 100_2 \\ [1,0,1] &\leftrightarrow 101_2 \\ [1,1,0] &\leftrightarrow 110_2 \\ [1,1,1] &\leftrightarrow 111_2 \end{aligned}$$

Jedem dieser binär codierten 3-Tupel wird nun der dem Regelsatz entsprechende Zustand aus der Zustandsmenge zugeordnet:

$$\begin{aligned} [0,0,0] &\leftrightarrow 000_2 \rightarrow 0 \\ [0,0,1] &\leftrightarrow 001_2 \rightarrow 1 \\ [0,1,0] &\leftrightarrow 010_2 \rightarrow 1 \\ [0,1,1] &\leftrightarrow 011_2 \rightarrow 1 \\ [1,0,0] &\leftrightarrow 100_2 \rightarrow 0 \\ [1,0,1] &\leftrightarrow 101_2 \rightarrow 1 \\ [1,1,0] &\leftrightarrow 110_2 \rightarrow 1 \\ [1,1,1] &\leftrightarrow 111_2 \rightarrow 0 \end{aligned}$$

Schreibt man diese acht Zahlen nun nebeneinander auf, indem man noch die Reihenfolge umkehrt (d.h. man beginnt mit dem Zustandswert, der der größten codierenden Binärzahl entspricht), so ergibt sich die folgende Anordnung:

01101110

Diese Zahl lässt sich wiederum als binär codierte Zahl

01101110_2

auffassen, die man nun noch in einem letzten Schritt dezimal codiert:

$$01101110_2 = 110_{10} = 110 \quad (\text{bei dezimal codierten Zahlen wird der Index üblicherweise weggelassen}).$$

Damit wird also der Beispielautomat durch die Bezeichnung eZA_1N2Z_110 eindeutig spezifiziert. Diese Art der Notation lässt sich auf beliebige z- und n-Werte verallgemeinern. Die Notation ist eineindeutig, d.h. es lässt sich aus der Automatenbezeichnung neben dem Automatentyp auch der Regelsatz eindeutig erschließen. Dies soll anhand eines weiteren Beispiels illustriert werden:

Es sei der Automat mit der Bezeichnung eZA_1N3Z_2047127316893 gegeben.

Da der ZA die Zustandsanzahl $z=3$ besitzt, muss der dezimal codierte Regelsatz in eine Zahl zur Basis 3 transformiert werden. Dies ergibt wegen $2047127316893_{10} = 21020200222211012121012022_3$ den Regelsatzcode $21020200222211012121012022_3$. Da es wegen $n=1$ zu einer Zelle insgesamt 3 Nachbarzellen gibt, existieren $3^3 = 27$ Nachbarschaftszustandskonfigurationen und somit 27 verschiedene 3-Tupel, denen der Regelsatz Zahlen aus der Zustandsmenge $Z = \{0, 1, 2\}$ zuordnet. Nun besitzt die obige im Dreier-System dargestellte Zahl $21020200222211012121012022_3$ jedoch nur 26 Ziffern. Das liegt daran, dass gemäß Konvention führende Nullen nicht aufgeschrieben werden. Die Zahl muss also um eine führende Null ergänzt werden, so dass sich dann $021020200222211012121012022_3$ ergibt. Daraus lässt sich dann anschließend der Regelsatz eindeutig ableiten:

$f([0, 0, 0])=2, f([0, 0, 1])=2, f([0, 0, 2])=0,$
 $f([0, 1, 0])=2, f([0, 1, 1])=1, f([0, 1, 2])=0,$
 $f([0, 2, 0])=1, f([0, 2, 1])=2, f([0, 2, 2])=1,$
 $f([1, 0, 0])=2, f([1, 0, 1])=1, f([1, 0, 2])=0,$
 $f([1, 1, 0])=1, f([1, 1, 1])=1, f([1, 1, 2])=2,$
 $f([1, 2, 0])=2, f([1, 2, 1])=2, f([1, 2, 2])=2,$
 $f([2, 0, 0])=0, f([2, 0, 1])=0, f([2, 0, 2])=2,$
 $f([2, 1, 0])=0, f([2, 1, 1])=2, f([2, 1, 2])=0,$
 $f([2, 2, 0])=1, f([2, 2, 1])=2, f([2, 2, 2])=0.$

Belegt man eZA_1N3Z_2047127316893 bei einer Weltbreite von $wb=200$ mit einer zufälligen Startbelegung, so ergibt sich für die ersten 200 Zellgenerationen beispielsweise die folgende Raum-Zeit-Entfaltung:

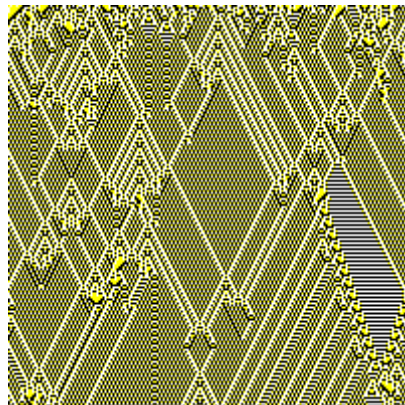


Abb.1.6

Der in der Einleitung angedeutete Klassifikationsalgorithmus arbeitet vor allen Dingen bei eZA mit symmetrischem Regelsatz recht zuverlässig. Das sind solche Automaten, für die gilt:

$f([z_0, z_1, \dots, z_{n-1}, z_n, z_{n+1}, \dots, z_{n+n}]) = f([z_{n+n}, z_{n+n-1}, \dots, z_{n+1}, z_n, z_{n-1}, \dots, z_0])$, wobei z_0, z_1, \dots, z_{n+n} beliebige Zahlen aus der Zustandsmenge Z sind.

Salopp ausgedrückt sind eZA mit symmetrischem Regelsatz Automaten mit der Eigenschaft:

$f([\text{links}, z_{\text{Mitte}}, \text{rechts}]) = f([\text{rechts}, z_{\text{Mitte}}, \text{links}])$.

Im Folgenden werden eZA mit symmetrischem Regelsatz mit esZA abgekürzt.

In der Literatur werden häufig sogenannte totalistische ZA betrachtet. Bei diesen Automaten hängt der Zellzustand einer Zelle in der nächsten Generation von der Summe der Zustände ihrer Nachbarzellen in der jetzigen Generation ab. Damit ist die Menge der totalistischen eZA eine Teilmenge der esZA, und alle Aussagen für esZA gelten insbesondere für totalistische eZA !

Bei dem ersten der obigen zwei Beispiele für eZA, also bei eZA_1N2Z_110, handelt es sich nicht um einen esZA, da $f([0,0,1])=1$ verschieden von $f([1,0,0])=0$ ist. Hingegen besitzt der Automat eZA_1N3Z_2047127316893 einen symmetrischen Regelsatz, so dass hier die Bezeichnung esZA_1N3Z_2047127316893 angemessen ist.

Um Untersuchungsergebnisse zu eZA und esZA besser einschätzen zu können, ist es immer wieder wichtig zu wissen, wie groß die Anzahl der Automaten zu einem vorgegebenen Automatentyp eZA_nNzZ bzw. esZA_nNzZ ist. Dazu die folgenden Überlegungen:

Hat ein eZA $2 \cdot n + 1$ Nachbarzellen und z Zustände, so gibt es $z^{2 \cdot n + 1}$ verschiedene Nachbarschaftszustandskonfigurationen und somit $z^{2 \cdot n + 1}$ verschiedene $(2 \cdot n + 1)$ -Tupel des Regelsatzes, da für jede Koordinate des Tupels z Möglichkeiten der Festlegung existieren. Da es für jedes dieser $(2 \cdot n + 1)$ -Tupel

z Möglichkeiten der Zuordnung für den Regelsatz gibt, existieren also insgesamt $z^{2 \cdot n + 1}$ verschiedene Regelsätze.

Also existieren $z^{2 \cdot n + 1}$ verschiedene Automaten des Typs eZA_nNzZ.

Die Anzahl der Automaten des Typs esZA_nNzZ zu bestimmen ist etwas aufwändiger. Man betrachtet dazu wie schon bei den eZA die $(2 \cdot n + 1)$ -Tupel der möglichen Nachbarschaftszustandskonfigurationen, also die $(2 \cdot n + 1)$ -Tupel, denen der Regelsatz des Automaten Elemente aus der Menge Z zuordnet:

Zunächst gibt es $z^{2 \cdot n + 1}$ verschiedene $(2 \cdot n + 1)$ -Tupel, die symmetrisch bezüglich ihrer mittleren Koordinate n sind. Für jedes dieser $z^{2 \cdot n + 1}$ verschiedenen $(2 \cdot n + 1)$ -Tupel gibt es z Möglichkeiten der Zuordnung beim Regelsatz.

Da es insgesamt $z^{2 \cdot n + 1}$ verschiedene $(2 \cdot n + 1)$ -Tupel gibt, ist $z^{2 \cdot n + 1} - z^{n + 1}$ die Anzahl der nicht zu ihrer mittleren Koordinate symmetrischen $(2 \cdot n + 1)$ -Tupel. Da der Regelsatz symmetrisch sein soll, werden jeweils zweien dieser insgesamt $z^{2 \cdot n + 1} - z^{n + 1}$ verschiedenen $(2 \cdot n + 1)$ -Tupel die gleichen Zahlen aus Z zugeordnet. Ordnet man nämlich einem dieser $(2 \cdot n + 1)$ -Tupel ein Element aus Z zu, so erhält das zugehörige $(2 \cdot n + 1)$ -Tupel, das durch Spiegelung an der mittleren Koordinate entsteht und von dem ersten Tupel ja verschieden ist, den gleichen Wert aus Z . Es sind also effektiv nur $\frac{1}{2} \cdot (z^{2 \cdot n + 1} - z^{n + 1})$ verschiedene $(2 \cdot n + 1)$ -Tupel für die Zuordnung beim Regelsatz zu berücksichtigen. Für jedes dieser $\frac{1}{2} \cdot (z^{2 \cdot n + 1} - z^{n + 1})$ verschiedenen $(2 \cdot n + 1)$ -Tupel gibt es wieder z Möglichkeiten der Zuordnung beim Regelsatz.

Insgesamt existieren damit

$$\binom{1}{z} \cdot \binom{z^{2 \cdot n + 1} - z^{n + 1}}{z^{n + 1}} + z^{n + 1} = \binom{1}{z} \cdot \binom{z^{2 \cdot n + 1} + z^{n + 1}}{z^{n + 1}} = \binom{1}{z} \cdot z^{n + 1} \cdot \binom{z^{n + 1}}{z^{n + 1}}$$

verschiedene Regelsätze und damit Automaten vom Typ esZA_nNzZ .

Die folgende kleine Tabelle Tab.1.1 zeigt einige konkrete Zahlen:

n	z	Anzahl der Automaten vom Typ eZA_nNzZ	Anzahl der Automaten vom Typ esZA_nNzZ
1	2	256	64
1	3	7 625 597 484 987	387 420 489
1	4	340 282 366 920 938 463 463 374 607 431 768 211 456 $\approx 3,40282 \cdot 10^{38}$	1 208 925 819 614 629 174 706 176 $\approx 1,20893 \cdot 10^{24}$
1	5	$\approx 2,35099 \cdot 10^{87}$	$\approx 2,64698 \cdot 10^{52}$
1	6	$\approx 1,20412 \cdot 10^{168}$	$\approx 1,11444 \cdot 10^{98}$
1	7	$\approx 7,38972 \cdot 10^{289}$	$\approx 4,35728 \cdot 10^{165}$
1	8	$\approx 2,41031 \cdot 10^{462}$	$\approx 1,23003 \cdot 10^{260}$
1	9	$\approx 4,39329 \cdot 10^{695}$	$\approx 2,93911 \cdot 10^{386}$
2	2	4 294 967 296	1 048 576
2	3	$\approx 8,71896 \cdot 10^{115}$	$\approx 2,57851 \cdot 10^{64}$
2	4	$\approx 3,2317 \cdot 10^{616}$	$\approx 3,31616 \cdot 10^{327}$
3	2	$\approx 3,40282 \cdot 10^{38}$	$\approx 4,72237 \cdot 10^{21}$
4	2	$\approx 1,34078 \cdot 10^{154}$	$\approx 7,58855 \cdot 10^{81}$
5	2	$\approx 3,2317 \cdot 10^{616}$	$\approx 7,72103 \cdot 10^{317}$

Tab.1.1

In der Einleitung wurden die vier Wolframschen Automatenklassen schon beschrieben. Diese Beschreibung soll an dieser Stelle nicht weiter vertieft werden. Es wird lediglich noch einmal auf die in der Einleitung erwähnte Literatur verwiesen. Allerdings sollen nachfolgend jeweils zwei Beispiele für Raum-Zeit-Entfaltungen von esZA der Klassen I bis IV gezeigt werden.



Abb.1.7

Abb.1.7 zeigt den Klasse I - Automaten esZA_1N2Z_250.

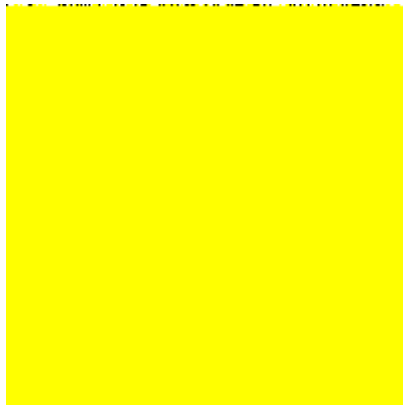


Abb.1.8

Abb.1.8 zeigt den Klasse I - Automaten

esZA_2N3Z_4359482124298047910145553529293038588484820362023658750427626097189954835468
61719971737774953415841558395527612832813.

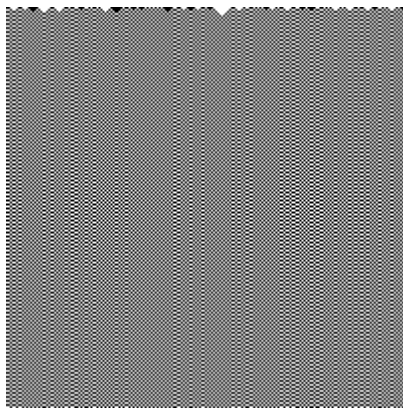


Abb.1.9

Abb.1.9 zeigt den Klasse II - Automaten esZA_1N2Z_178.

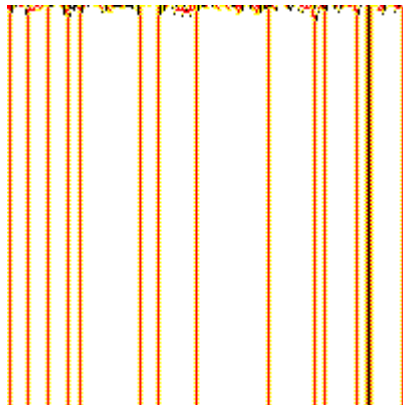


Abb.1.10

Abb.1.10 zeigt den Klasse II - Automaten esZA_1N4Z_26446168315790732842108823809055654672 .

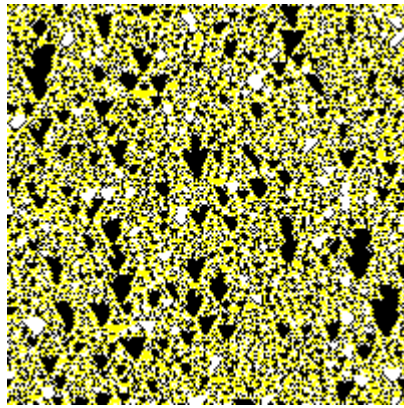


Abb.1.11

Abb.1.11 zeigt den Klasse III - Automaten

esZA_2N3Z_8489819411851162880654161093039115582695250937409948939972257430212357042790
3215941430353001136454268075889544637714.

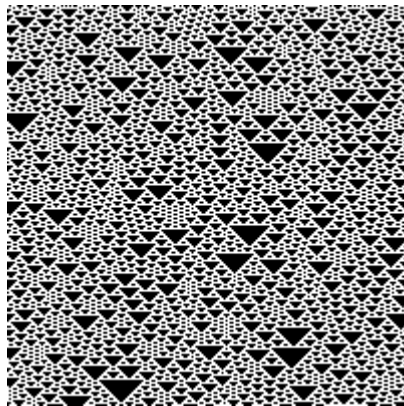


Abb.1.12

Abb.1.12 zeigt den Klasse III - Automaten esZA_1N2Z_129.

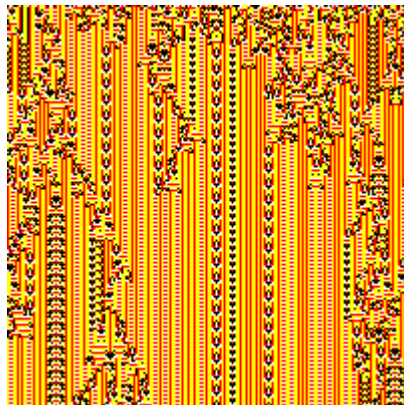


Abb.1.13

Abb.1.13 zeigt den Klasse IV - Automaten esZA_1N4Z_277690326185219388953999620655961249386.

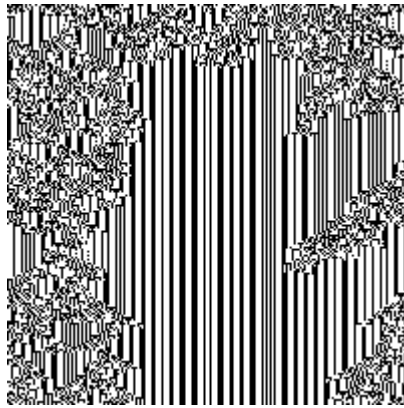


Abb.1.14

Abb.1.14 zeigt den Klasse IV - Automaten esZA_3N2Z_272848628671011874400214955638274447209.

2. Der Klassifikationsalgorithmus

Der im Folgenden vorgestellte Klassifikationsalgorithmus beruht auf einer speziellen Form der Datenkompression:

Ist

$$C(t) = [c_0(t) , c_1(t) , c_2(t) , \dots , c_{wb-2}(t) , c_{wb-1}(t)]$$

die Zellzustandskonfiguration eines eZA zu einem gegebenen Zeitpunkt t, so lässt sich C(t) eindeutig durch die Zahl

$$C_S(t) := \sum_{k=0}^{wb-1} c_k(t) \cdot z^k$$

kodieren. Durch die Bezeichnung $C_S(t)$ soll die Summation angedeutet werden. Da die größte so gebildete Zahl z^{wb-1} sehr groß sein kann, ist es zweckmäßig, $C_S(t)$ durch die folgende Transformation auf natürliche Zahlen im Intervall $[0; N_{\max}[$ einzuschränken:

$$rkC(t) := \text{Int} \left(N_{\max} \cdot \frac{\sum_{k=0}^{wb-1} c_k(t) \cdot z^k}{z^{wb}} \right)$$

wobei die Funktion $\text{Int}(r)$ den ganzzahligen Anteil einer reellen Zahl r ermittelt.

Die Bezeichnung $rkC(t)$ wurde gewählt, weil C(t) durch die Transformation gewissermaßen *räumlich komprimiert* wird.

Mit der Kompression ist i.a. ein enormer Datenverlust verbunden, da viele Zellzustandskonfigurationen C(t) den gleichen Wert $rkC(t)$ im Intervall $[0; N_{\max}[$ besitzen. Je kleiner N_{\max} ist, desto größer ist der Datenverlust. Wie sich herausstellt, beeinträchtigt dieser Datenverlust die weiteren Analysen jedoch nicht, falls N_{\max} nicht zu klein gewählt wird. Im Gegenteil: gerade durch den Datenverlust werden Muster

sichtbar, welche nachfolgend den Klassifikationsalgorithmus überhaupt erst ermöglichen.

Durch die Einführung der räumlichen Kompression rkC des eZA wird aus dem zweidimensionalen Problem ein eindimensionales. An die Stelle der zweidimensionalen Raum-Zeit-Entfaltung tritt die eindimensionale Zeitentfaltung von rkC. Diese Entfaltung soll im Folgenden räumlich komprimierte Zeitentfaltung des ZA (kurz: rkZ) genannt werden.

Entscheidend ist nun, dass sich die rkZ der vier Automatenklassen (in den allermeisten Fällen!) in spezifischer Weise voneinander unterscheiden. Anhand der nachfolgenden Beispiele wird dies deutlich. Es wurde jeweils $N_{\max} = 5000$ gewählt.

Die ersten zwei Abbildungen Abb.2.1 und Abb.2.2 zeigen die ersten 200 Generationen der rkZ eines Automaten der Klasse III (esZA_2N2Z_1069308648), jeweils ausgehend von einer zufälligen Startbelegung:

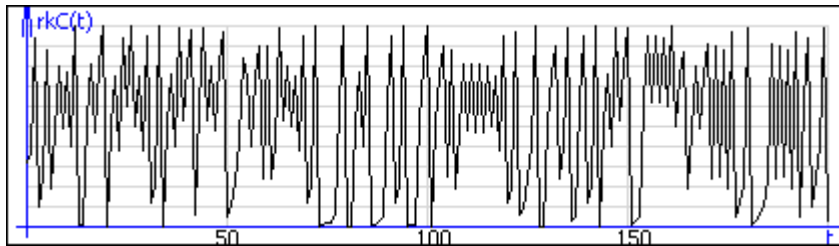


Abb.2.1

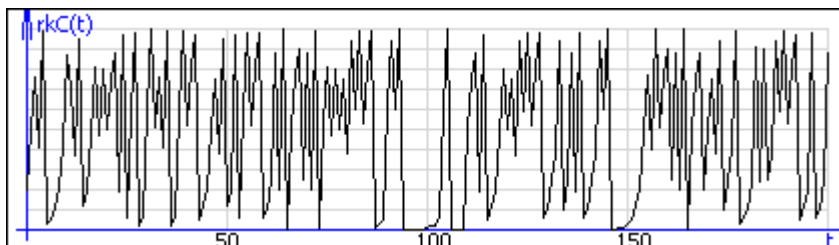


Abb.2.2

In der Abbildung Abb.2.3 ist zum Vergleich die vertraute Raum-Zeit-Entfaltung des Automaten dargestellt:

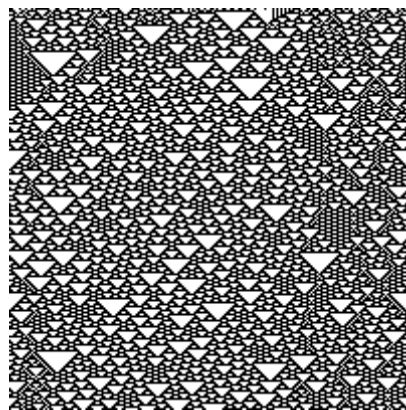


Abb.2.3

Man erkennt, dass die rkC(t)-Werte erratisch schwanken. In der rkZ ist keine Ordnung ersichtlich. Dieses Fehlen von offensichtlichen Ordnungsstrukturen in der rkZ ist ein typisches Kennzeichen von Klasse III - Automaten und wird immer wieder beobachtet.

Anders liegen die Verhältnisse bei Klasse IV - Automaten. Die nachfolgende Abbildung 2.4 zeigt zunächst eine Raum-Zeit-Entfaltung des Klasse IV - Automaten

esZA_2N3Z_86101018441089936901491278601835372126880646949764121042859102773692712091439773043952501672248340502916745135354145:

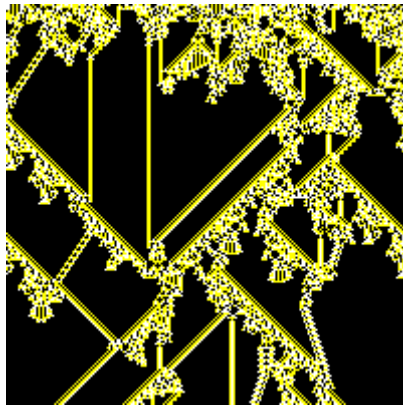


Abb.2.4

Nachfolgend wie oben die ersten 200 Generationen zweier rkZ des esZA bei jeweils zufälliger Startbelegung:

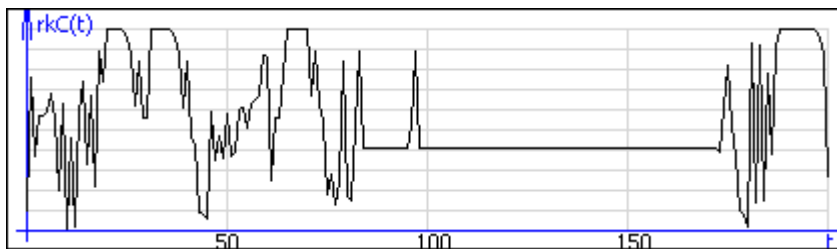


Abb.2.5

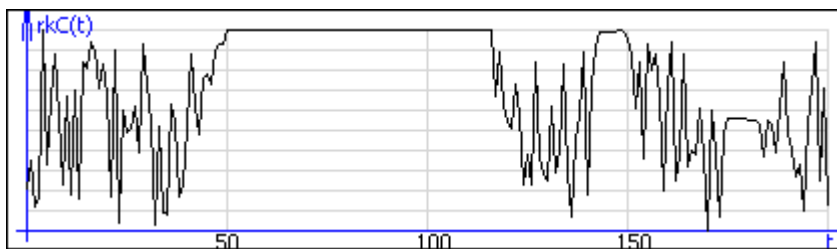


Abb.2.6

Es sind hier zwar auch erratische Schwankungen zu beobachten, jedoch nicht, und das ist entscheidend, durchgängig. Phasen wilder und unvorhersehbarer Schwankungen werden unterbrochen von Phasen mit nur wenigen $rkC(t)$ -Werten. Diese geordneten Phasen können 40 und mehr Zeitschritte lang sein. Speziell bei diesem Automaten treten sie immer wieder auf. Es ist nicht vorhersehbar, wann sie jeweils einsetzen und wann sie enden. Der Grund für diese Phasen annähernder Konstanz der $rkC(t)$ -Werte liegt auf der Hand: bei einem Automaten der Klasse IV entwickeln sich immer wieder mehr oder weniger ausgedehnte Gebiete mit gleichem oder räumlich periodischem Zellzustand. Diese Gebiete bleiben für einen begrenzten Zeitraum erhalten. Liegen die Gebiete in dem Bereich des esZA, der bei der Transformation von $C(t)$ nach $C_s(t)$ in hohe Potenzen von j überführt wird, so liegen zu aufeinanderfolgenden Zeitpunkten die Zahlen $C_s(t)$, $C_s(t+1)$, $C_s(t+2)$,... relativ nahe beieinander, da ja die Bereiche des Automaten, in denen von Zeitschritt zu Zeitschritt überall Änderungen der Zellzustände auftreten, durch ihre Kodierung mit kleinen Potenzen nur relativ wenig zur Größe der Zahl $rkC(t)$ beitragen. Dies führt dazu, dass die Zahlen $rkC(t)$, $rkC(t+1)$, $rkC(t+2)$,... über einen begrenzten Zeitraum identisch sind, was die Phasen der zeitlichen Konstanz im $rkC(t)$ -t-Diagramm erklärt.

Ein Beispiel soll dies verdeutlichen:

Es werde ein esZA_1N4Z mit 100 Zellen betrachtet. Die Zellen von Nr.0 bis Nr.82 sollen beliebig belegt sein, die Zellen von Nr. 83 bis 97 mit Zustand 0, die Zelle Nr.98 mit Zustand 2 und die Zelle Nr.99 mit Zustand 0 (d.h. im Zellraum des Automaten befindet sich ein Gebiet der Länge 15 mit einheitlichem Zellzustand 0). Da die Zellen von Nr.0 bis Nr.82 beliebig belegt sein können, gilt für die Zahl $C_s(t)$:

$$2,00867255532 \cdot 10^{59} \approx 2 \cdot 4^{98} \leq C_s(t) \leq 1 \cdot 4^{83} + 2 \cdot 4^{98} \approx 2,00867255626 \cdot 10^{59} .$$

Mit $N_{\max} = 1000$ ist $rkC(t) = \text{Int}(1000 \cdot Cs(t)/4^{100}) = 125$, und zwar unabhängig von den Zellzuständen in den ersten 83 Zellen.

Es soll nun angenommen werden, dass sich das Gebiet im darauffolgenden Zeitschritt links um eine Zelle verkleinert und rechts gleich bleibt (d.h. $c_{84}(t+1) = c_{85}(t+1) = \dots = c_{97}(t+1) = 0$, $c_{98}(t+1) = 2$, $c_{99}(t+1) = 0$). Da die Zellen von Nr.0 bis Nr.83 beliebig belegt sein können, gilt für die Zahl $Cs(t+1)$:

$$2,00867255532 \cdot 10^{59} \approx 2 \cdot 4^{98} \leq Cs(t+1) \leq 1 \cdot 4^{84} + 2 \cdot 4^{98} \approx 2,00867255907 \cdot 10^{59}.$$

Damit hat $rkC(t+1)$ ebenfalls den Wert 125, völlig unabhängig von den Zellzuständen in den ersten 84 Zellen!

Liegen im esZA die Gebiete homogenen Zellzustandes auf der Seite, die mit niedrigen Exponenten von j kodiert wird, so tritt der beobachtete Effekt natürlich nicht auf. Da aber bei zufälliger Startbelegung die Gebiete mit mehr oder weniger einheitlichem Zellzustand früher oder später überall im Zellraum auftreten, kommt es immer wieder zu den Phasen zeitlicher Konstanz im $rkC(t)$ -t-Diagramm.

Ein anderes typisches Muster der rkZ bei Klasse IV - Automaten wird beispielhaft sichtbar bei esZA_1N4Z_69029527746430263732297465511054586065. Die nachfolgende Abbildung 2.7 zeigt eine Raum-Zeit-Entfaltung dieses Automaten:

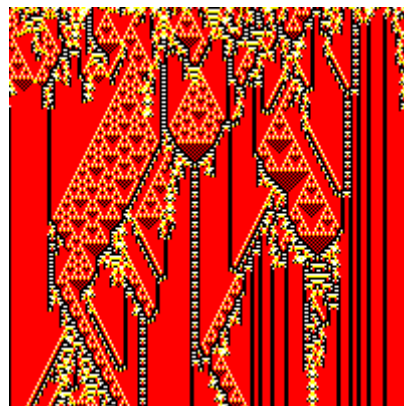


Abb.2.7

Nachfolgend wieder zwei $rkC(t)$ -t-Diagramme, jeweils ausgehend von zufälliger Startbelegung:

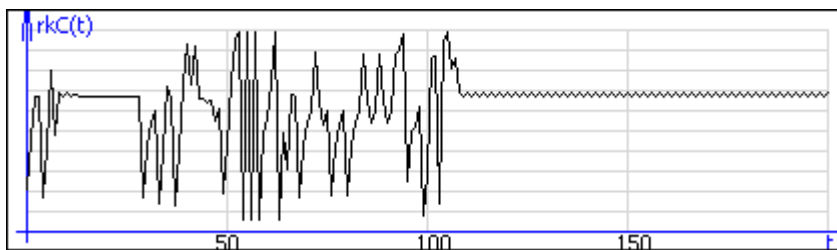


Abb.2.8

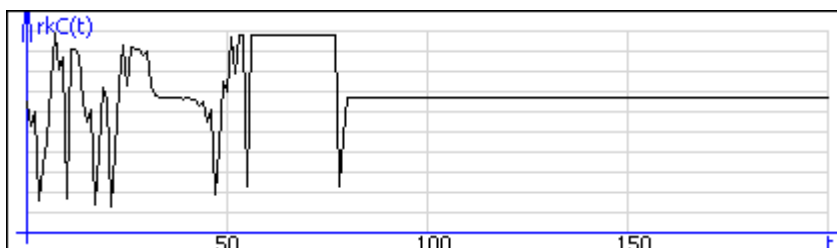


Abb.2.9

Nach anfänglich erraticem Verhalten werden die $rkC(t)$ -Werte nach relativ kurzer Zeit stabil bzw. periodisch mit recht kurzer Periodendauer. Die oben beschriebenen mehr oder weniger langen Ordnungsphasen inmitten von Phasen mit unvorhersagbarem Verhalten treten eigentlich nicht auf. Dies

rührt daher, dass der Automat nach ca. 100 Zeitschritten in ein stationäres Muster übergeht, das an einen Klasse II-Automaten erinnert: einfaches Muster mit kurzer Periodendauer.

Einfache Strukturen der ekZ besitzen Automaten der Klasse II, wie das nachfolgende Beispiel (esZA_1N2Z_91) zeigt. Zunächst eine Raum-Zeit-Entfaltung:

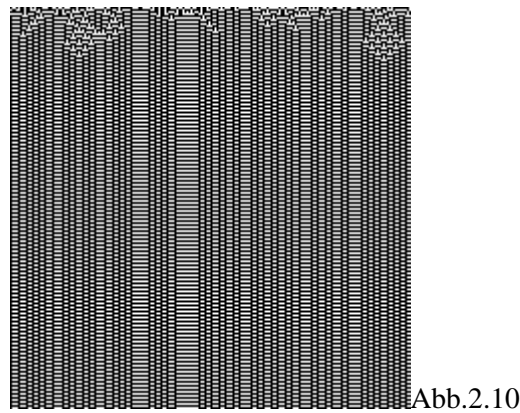


Abb.2.10

Anschließend drei verschiedene rkZ:

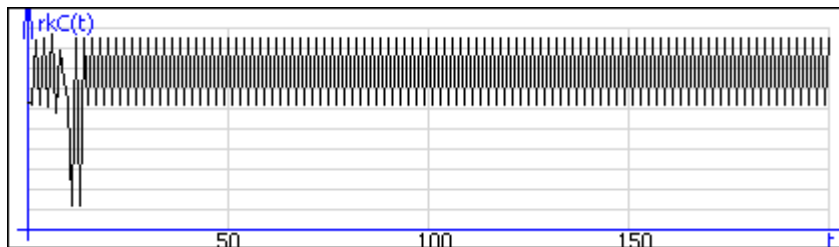


Abb.2.11

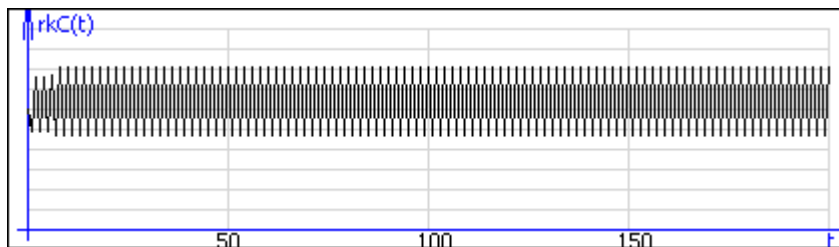


Abb.2.12

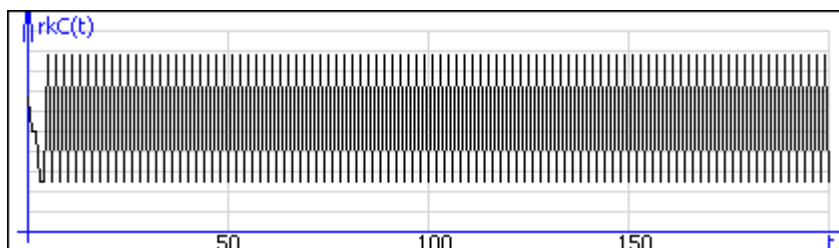


Abb.2.13

Nach einer sehr kurzen Einschwingzeit werden die $rkC(t)$ -Werte periodisch mit sehr kurzer Periodendauer oder häufig sogar konstant, wie das nachfolgende Beispiel eines weiteren Klasse II - Automaten zeigt. Zuerst wieder die Raum-Zeit-Entfaltung des Automaten (esZA_1N4Z_92296382473385718170978943278039669603):

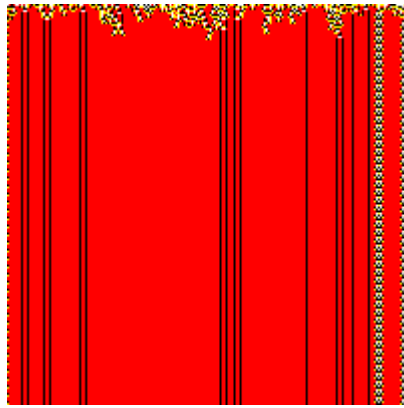


Abb.2.14



Abb.2.15

Diese Struktur der ekZ erinnert stark an die rkZ des oben angesprochenen Klasse IV - Automaten esZA_1N4Z_69029527746430263732297465511054586065 (siehe Abb.2.9), mit dem Unterschied, dass die zeitliche Konstanz der rkC(t)-Werte bei dem Klasse II - Automaten sehr viel früher einsetzt als beim Klasse IV - Automaten. Der Klassifikationsalgorithmus hat dies zu berücksichtigen.

Es ist klar, dass die rkZ eines Klasse I - Automaten ähnlich der eines Klasse II - Automaten ist. Das folgende Beispiel (esZA_2N2Z_4291755721) zeigt dies:



Abb.2.16

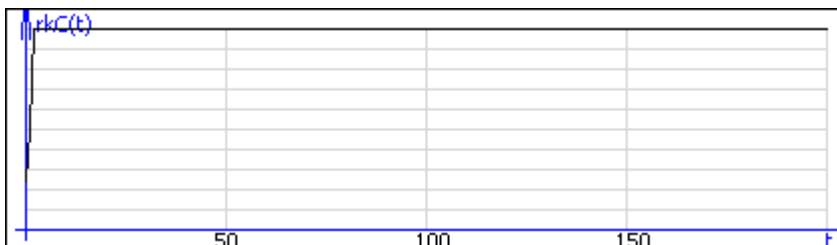


Abb.2.17

Bei einem Klasse I - Automaten befinden sich jedoch im Gegensatz zu Klasse II - Automaten alle Zellen nach kurzer Einschwingzeit im gleichen Zellzustand. Der Klassifikationsalgorithmus verwendet diesen Unterschied zur Differenzierung zwischen Klasse II - und Klasse I - Automaten.

Das bisher skizzierte Bild bedarf noch der Ausschärfung, da es einige besondere Klasse II - Automaten gibt, die mit Hilfe des bisher beschriebenen Rasters nicht klar genug identifiziert werden können. Der Automat esZA_2N3Z_74858973544172052105042518917251377268586316442320480123336345658129942743611154699714930654682600242364728913638238 ist solch ein Beispiel. Anfangs wieder die Raum-Zeit-Entfaltung:

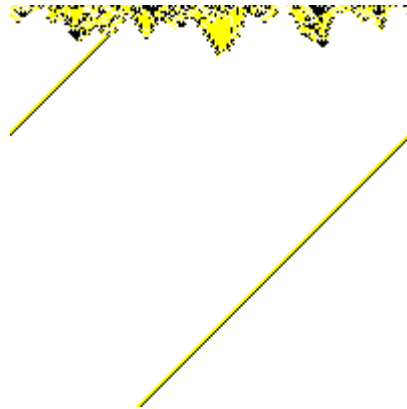


Abb.2.18

Natürlich gehört dieser Automat weder zur Klasse IV noch zur Klasse III, da seine Entfaltung viel zu einfach strukturiert ist. Auch die Einstufung als Klasse I-Automat ist nicht gegeben, weil er nie einen homogenen Zellzustand erreicht. Bleibt nur die Einordnung in die Klasse II. Aber Klasse II-Automaten waren dadurch gekennzeichnet, dass sie nach wenigen Zeitschritten einen periodischen Zustand geringer Periodendauer erreichen. Bei dem Automaten aus Abb.2.18 beträgt die Weltbreite 200. Aus der Abbildung wird deutlich, dass nach weniger als 200 Zellgenerationen ein periodischer Zustand erreicht wird. Eine Störung wandert dann mit konstanter Geschwindigkeit (pro Zeitschritt eine Zelle) nach links über den Zellbereich. Wegen der zyklischen Zellstruktur des Automaten setzt sie, wenn sie am linken Rand angekommen sind, ihre gleichförmige Bewegung vom rechten Rand aus beginnend fort, wandert zum linken Rand, usw. ad infinitum, falls man den esZA nicht anhält. Somit beträgt die Periodendauer 200 Zeitschritte, da eine solche Störung in einer Zeiteinheit eine Längeneinheit zurücklegt und die Zellenanzahl 200 beträgt. Diese Periodendauer kann man nicht mehr kurz nennen. Außerdem hängt sie von der Zellenanzahl ab. Besäße der Automat 1000 Zellen, so betrüge die Periodendauer entsprechend 1000 Zeitschritte. Bei einem Automaten mit unendlich vielen Zellen wäre die Periodendauer sogar unendlich! Doch trotz dieser Diskrepanz zwischen Entfaltungseigenart des Automaten und der begrifflichen Festlegung kommt eigentlich nur die Einordnung des Automaten in die Klasse II in Frage, denn ohne Zweifel ist sein Raum-Zeit-Muster sehr ähnlich zu den bisher betrachteten Klasse II-Automaten. Um das einzusehen, braucht man nur einmal in Gedanken nach jedem Zeitschritt alle Zellen um eine Zelle zyklisch nach rechts zu verschieben und dann die neue Zellgeneration zu erzeugen. Dann entsteht ein Klasse II-Automat par excellence mit kurzer Periodendauer 1.

Die ersten 400 Schritte der rkZ des Automaten sehen beispielsweise so aus:

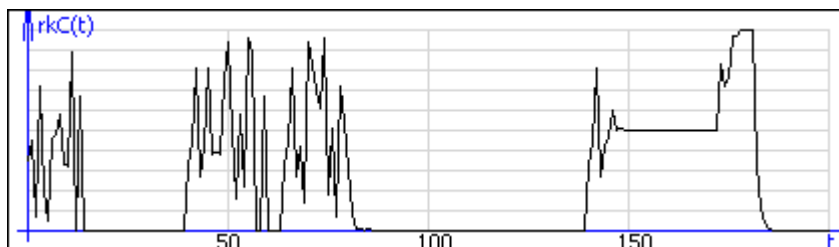


Abb.2.19

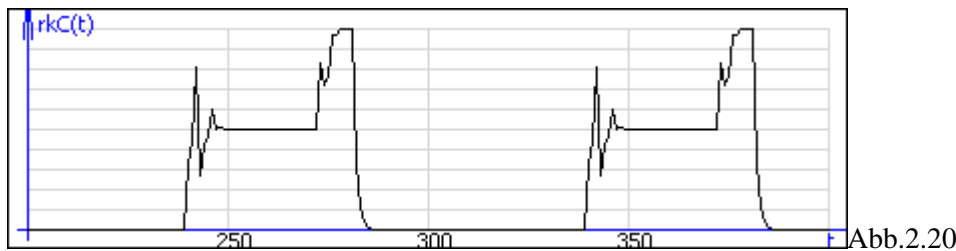


Abb.2.20

Phasen annähernder zeitlicher Konstanz werden regelmäßig unterbrochen von kurzen Perioden der Schwankung. Das liegt daran, dass sich die gerade beschriebenen Störungen mit konstanter Geschwindigkeit zyklisch durch den Zellraum bewegen. Befinden sich diese Störungen im Bereich des Automaten, der bei der Berechnung von $C_s(t)$ mit kleinen Potenzen kodiert wird, so tragen sie so wenig zum $C_s(t)$ -Wert bei, so dass der $rkC(t)$ -Wert mit nachfolgenden $rkC(t)$ -Werten übereinstimmt (siehe oben). Erst wenn die Störungen in den Bereich des Automaten gelangen, der mit hohen Potenzen kodiert wird, wirken sie sich wahrnehmbar auf den $rkC(t)$ -Wert aus, der dann schwankt, und zwar so lange, bis die Störung sich wieder in den Teil des Zellraumes ausgebreitet hat, der mit kleinen Potenzen kodiert wird. Die rkZ dieses Automaten ähnelt stark der rkZ vieler Klasse IV - Automaten (siehe z.B. Abb.2.5 und Abb.2.6). Der Klassifikationsalgorithmus berücksichtigt dies, indem er überprüft, ob die Raum-Zeit-Entfaltung eine Periodendauer aufweist, die im Bereich der Weltbreite des esZA liegt, um Automaten wie den soeben betrachteten als Klasse II - Automaten zu identifizieren.

Mit dem Hilfsmittel der rkZ wird nun der Klassifikationsalgorithmus entworfen. Zunächst der Kernalgorithmus:

Kernalgorithmus

Zunächst wird, ausgehend von einer zufälligen Startbelegung des esZA, die rkZ des Automaten (mit einer Weltbreite von 100 bis 200 und $N_{max} = 5000$) über einen Zeitraum von 1000 Zeitschritten ermittelt, wobei sich der Wert 1000 für N_{max} als ausreichend groß erwiesen hat. Die Zahlen $rkC(0)$, $rkC(1)$, ..., $rkC(999)$ werden in einem Array abgelegt.

Als erstes wird überprüft, ob die rkZ eine zeitliche Periode der Dauer 2-Weltbreite oder 3-Weltbreite aufweist, und wenn, ab welchem Zeitpunkt sie beginnt. Diese Zahl soll Periodenstartzeitpunkt genannt werden.

Ist dieser Periodenstartzeitpunkt kleiner als 1000, so muss differenziert werden:

Ist der Periodenstartzeitpunkt kleiner als 30, so wird

ein Automat der Klasse I konstatiert, falls sich nach kurzer Einschwingzeit alle Zellen des Automaten im gleichen Zustand befinden,

ein Automat der Klasse II konstatiert, falls sich nach kurzer Einschwingzeit nicht alle Zellen des Automaten im gleichen Zustand befinden.

Ist der Periodenstartzeitpunkt größer als 30, so wird ein Automat der Klasse IV konstatiert.

Liegt hingegen keine Periode vor, so wird bei vorgegebener Zeitfensterlänge t_F (ein Wert für t_F von ca. 50 hat sich als ausreichend erwiesen) die minimale Anzahl von $rkC(t)$ -Werten innerhalb dieses Zeitfensters, und zwar bezogen auf das gesamte $rkC(t)$ -Array, ermittelt.

Ist diese minimale Anzahl höchstens 9, so wird ein Automat der Klasse IV konstatiert und ansonsten ein Automat der Klasse III.

Nun der Hauptalgorithmus:

Hauptalgorithmus

Der Kernalgorithmus wird bei einem gegebenen Automaten mehrere Male durchgeführt. Diejenige Klasse, die am häufigsten konstatiert wird, wird dann endgültig als Automatenklasse des Automaten ausgegeben. Dies ist notwendig, weil der Kernalgorithmus wegen der zufälligen Startbelegung nicht jedes Mal die gleiche Klasse ausgibt.

Es hat sich gezeigt, dass ein fünfzig- bis hundertmaliger Aufruf des Kernalgorithmus in den meisten Fällen ausreicht.

Die in der Programmiersprache C# geschriebene programmtechnische Realisierung des Klassifikationsalgorithmus ist im Anhang dargestellt.

3. Die Beurteilung des Klassifikationsalgorithmus

Die Güte des Klassifikationsalgorithmus richtet sich zunächst vor allem danach, wie häufig die subjektiven Klassifikationen zellulärer Automaten mit den algorithmischen übereinstimmen. Aus der Tabelle Tab.1.1 wird ersichtlich, dass die Codeanzahlen zu einem gegebenen Automatentyp esZA_nNzZ i.a. so hoch sind, dass an eine diesbezügliche systematische Durchmusterung aller Automaten eines Typs (bis auf wenige Ausnahmen) nicht zu denken ist. Man greift also strichprobenartig Automaten zu einem gegebenen Typ heraus und überprüft die Übereinstimmung zwischen subjektiver und algorithmischer Klassifikation. Aus programmieretechnischen Gründen wurde dieses Testverfahren lediglich auf esZA mit bis zu 3 Nachbarzellen pro Seite (d.h. $1 \leq n \leq 3$) und maximal 9 Zuständen (d.h. $2 \leq z \leq 9$) angewandt.

Der Automatentyp esZA_1N2Z:

So existieren lediglich 64 Automaten vom Typ esZA_1N2Z. Es zeigt sich bei diesem Automatentyp, dass subjektive und algorithmische Klassifikation zu 100% übereinstimmen.

Der Automatentyp esZA_2N2Z:

Beschränkt man sich zusätzlich auf totalistische esZA dieses Typs, so ist dies der (mittlerweile berühmte) Automatentyp, den Stephen Wolfram zur Grundlage seiner Einteilung zellulärer Automaten in die vier Klassen herangezogen hat (siehe [GeSch] S.72 - 86). Eine Stichprobe von 1000 zufällig ausgewählten Automaten ergab bei 998 Automaten Übereinstimmung zwischen algorithmischer und subjektiver Klassifikation. Die folgenden 6 Abbildungen zeigen entsprechende Raum-Zeit-Entfaltungen. In den unteren Teil des Bildes hat der Computer jeweils die von ihm ermittelte Automatenklasse eingetragen.

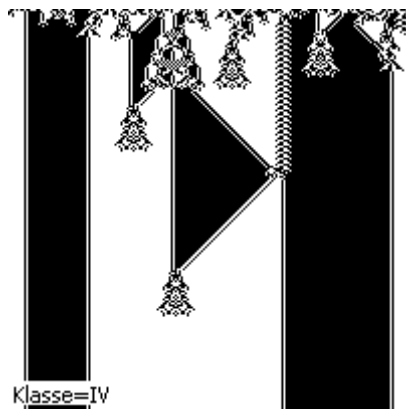


Abb.3.1 (esZA_2N2Z_3918960232)

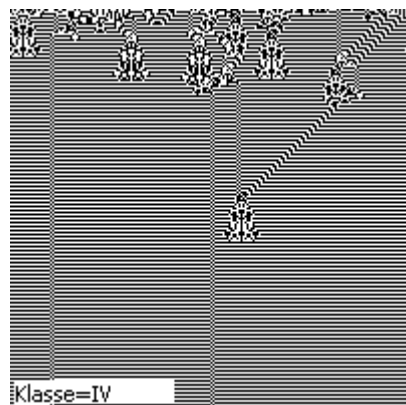


Abb.3.2 (esZA_2N2Z_376007063)

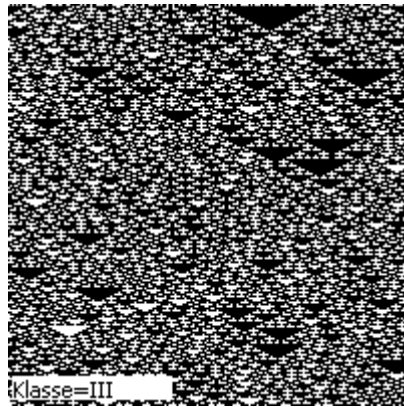


Abb.3.3 (esZA_2N2Z_2541715454)

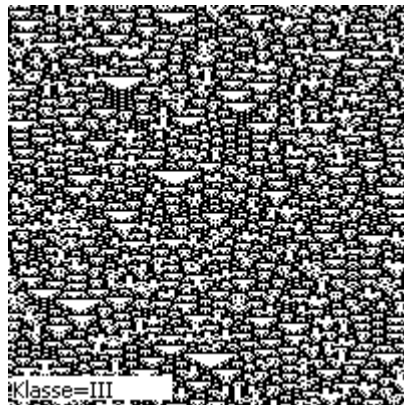


Abb.3.4 (esZA_2N2Z_2129258902)



Abb.3.5 (esZA_2N2Z_4276676736)

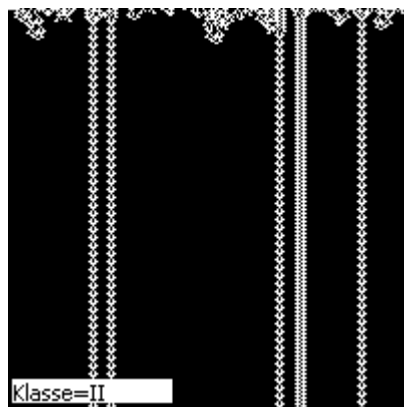


Abb.3.6 (esZA_2N2Z_3681084813)

Die zwei Automaten, bei denen subjektive und algorithmische Klassifikation nicht korrespondieren, sind esZA_2N2Z_964903050 und esZA_2N2Z_3747393469. Zunächst der Automat esZA_2N2Z_964903050:

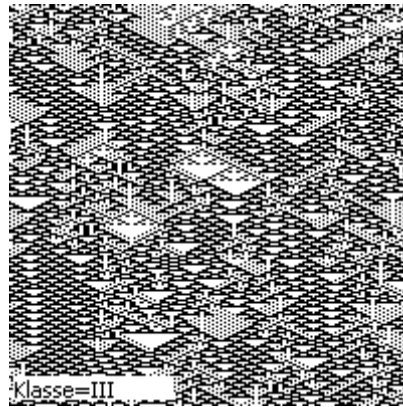


Abb.3.7

Aus der obigen Abbildung wird die unangemessene algorithmische Einstufung des Automaten noch nicht klar ersichtlich. Die folgende Grafik Abb.3.8 zeigt jedoch, dass der Automat der Klasse IV zuzuordnen ist, da neben eher chaotischen Bereichen deutliche Ordnungsstrukturen in der Raum-Zeit-Entfaltung erkennbar sind. Die Ordnungsstrukturen treten nicht sehr früh, sondern meistens erst nach über 1000 Zeitschritten auf.

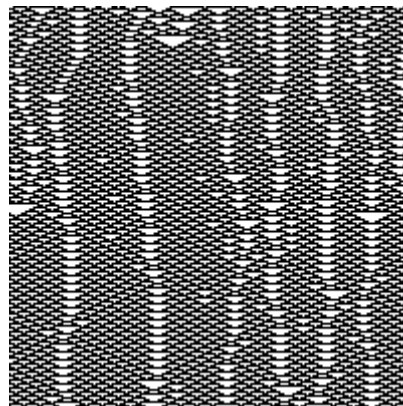


Abb.3.8

Bemerkenswert ist auch, wie der Klassifikationsalgorithmus zur Einordnung in die Klasse III gelangt: Bei 1000 Aufrufen des oben beschriebenen Kernalgorithmus wurde beispielsweise 559 Mal die Klasse III konstatiert und 441 Mal die Klasse IV, so dass der Hauptalgorithmus aufgrund seiner Konzeption endgültig die Klasse III ausgibt. Der Algorithmus erkennt also zwar ebenfalls in vielen Fällen die Klasse IV, entscheidet aber letztendlich wegen der Häufigkeit der getroffenen Einstufungen in die Klasse III unangemessen.

Der Automat esZA_2N2Z_3747393469 hat Raum-Zeit-Entfaltungen der folgenden Art:

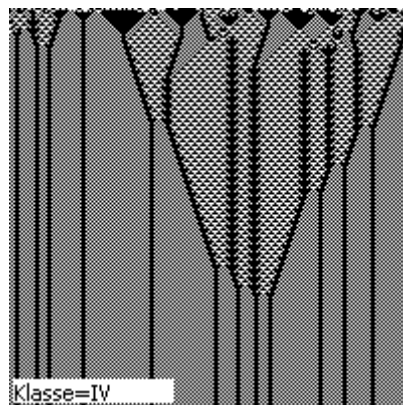


Abb.3.9

Wie man sofort sieht, ist die algorithmische Klassifikation unangemessen. Nach ca. 50 bis 150 Zeitschritten gelangt der Automat regelmäßig in eine zyklische Zustandskonfiguration mit geringer Periodendauer, wobei

die Raum-Zeit-Entfaltung bis zum Erreichen der zyklischen Zustandskonfiguration recht einfach verläuft und wenig Komplexität aufweist. Aus diesen Gründen ist die subjektive Einstufung eindeutig die in Klasse II. Der Kernalgorithmus muss jedoch aufgrund seiner Konzeption in den allermeisten Fällen hier die Klasse IV konstatieren. Das Beispiel einer rkZ verdeutlicht dies:

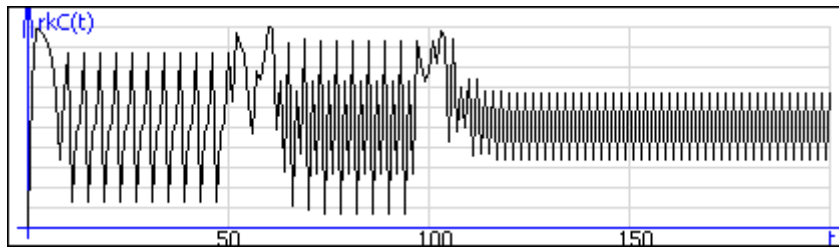


Abb.3.10

Solch eine rkZ ist eigentlich typisch für Klasse IV - Automaten, und entsprechend trifft der Klassifikationsalgorithmus seine Fehleinschätzung. Automaten dieser Art sind jedoch sehr selten, so dass der Algorithmus beibehalten wird.

Der Automatentyp esZA_3N2Z:

Hier ergab eine Stichprobe von 1000 Automaten bei 997 Automaten Übereinstimmung zwischen subjektiver und algorithmischer Klassifikation. Die drei Automaten mit fehlender Übereinstimmung waren:

esZA_3N2Z_274900756210629383173712957046794312081:

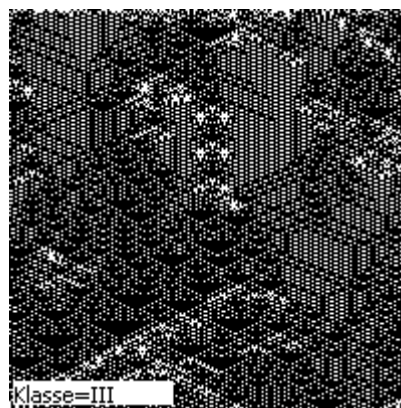


Abb.3.11

Bei 1000 Aufrufen des Kernalgorithmus ergab sich in 792 Fällen die Einordnung in die Klasse III und in 208 Fällen die Einordnung in die Klasse IV, also insgesamt eine unangemessene Klassifikation.

esZA_3N2Z_313479151769103633695467747196027186369:

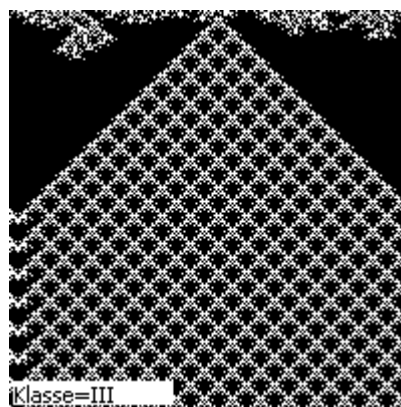


Abb.3.12

Die obige Raum-Zeit-Entfaltung zeigt, dass der Automat vom Algorithmus als zur Klasse III gehörig angesehen wird, dies aber offensichtlich nicht der Fall ist. Eher ist hier die Einstufung in die Klasse II

gegeben. Die rkZ zeigt den Grund für die unangemessene algorithmische Klassifikation:

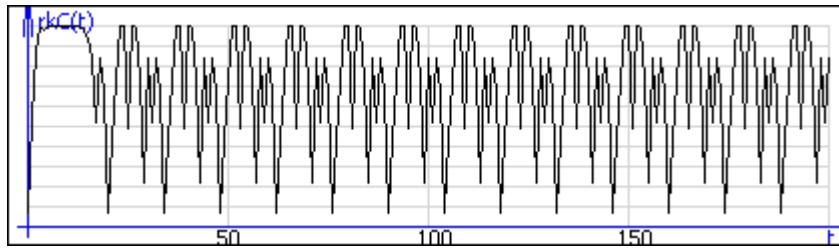


Abb.3.13

Die Zahl der angenommenen rkC(t)-Werte in dem vorgegebenen Zeitfenster (der Länge 50) ist so hoch, dass der Algorithmus hier auf Klasse III erkennt.

esZA_3N2Z_323954219910392711287182023698907977152:

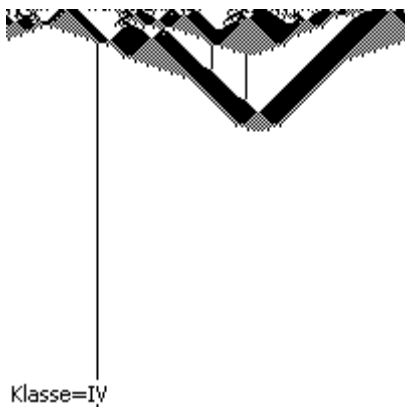


Abb.3.14

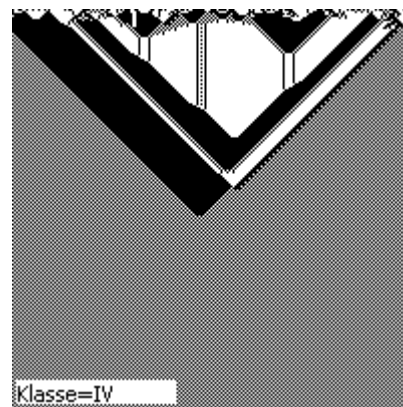


Abb.3.15

Die obigen zwei Raum-Zeit-Entfaltungen zeigen, dass die algorithmische Einordnung in Klasse IV wohl unangemessen ist. Wegen der zu geringen Komplexität der Raum-Zeit-Entfaltung in der Zeitspanne bis zum Erreichen des periodischen Zustands mit geringer Periodendauer ist eher die Einordnung in Klasse II erforderlich. Bei 1000 Aufrufen des Kernalgorithmus wurde beispielsweise 742 Mal auf Klasse IV erkannt und 258 Mal auf Klasse II, so dass der Hauptalgorithmus auf Klasse IV entschied.

Der Automatentyp esZA_1N3Z:

Auch bei diesem Automatentyp ergab sich ein sehr hoher Anteil übereinstimmender subjektiver und algorithmischer Klassifikationen. Lediglich bei 3 Automaten aus einer zufällig gewählten Stichprobe von 1000 mussten unangemessene algorithmische Klassifikationen festgestellt werden. Zwei dieser Beispiele sollen betrachtet werden: esZA_1N3Z_23665526179 und esZA_1N3Z_2859987108095:

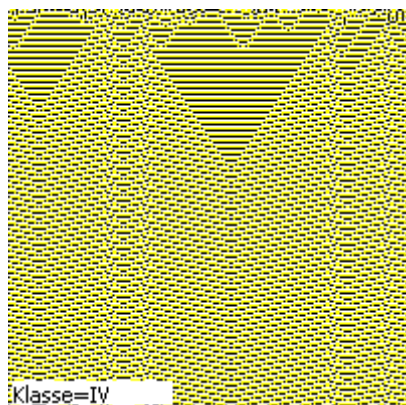


Abb.3.16(esZA_1N3Z_23665526179)

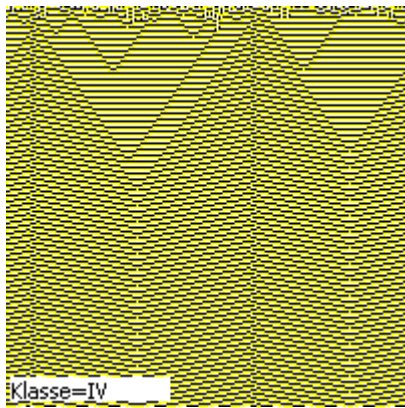


Abb.3.17(esZA_1N3Z_2859987108095)

Auf den ersten Blick ist bei diesen Automaten auch die subjektive Klassifikation unklar. Doch haben die Automaten die Eigenschaft, nach relativ wenigen Entfaltungen in einen Zyklus geringer Periodendauer überzugehen, so dass die subjektive Einstufung in die Klasse II sinnvoll erscheint. Der Klassifikationsalgorithmus konstatiert hingegen aufgrund seiner Programmierung die Klasse IV, weil in der rkZ in den allermeisten Fällen nach relativ kurzer Einschwingzeit von ca. 30 - 70 Zeitschritten ein Zyklus mit kurzer Periodendauer entsteht, wie es z.B. die nachfolgenden zwei Grafiken belegen:

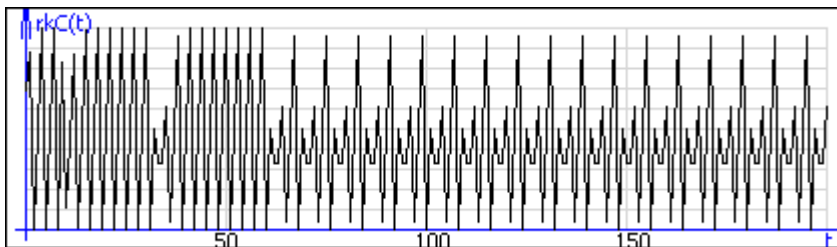


Abb.3.18

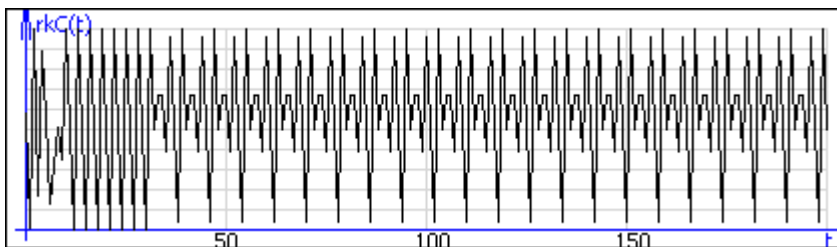


Abb.3.19

Der Automatentyp esZA_1N4Z:

Es ergab sich wie schon bei den oben betrachteten Automatentypen ein hoher Prozentsatz an Übereinstimmungen zwischen subjektiver und algorithmischer Klassifikation. Nur bei 2 von 1000 zufällig ausgewählten Automaten korrespondierten algorithmische und subjektive Klassifikation nicht:

esZA_1N4Z_203058938233882087919109080608130185052:

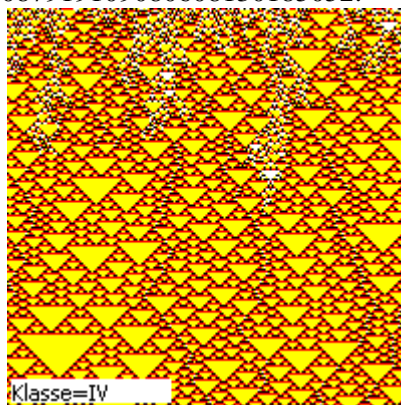
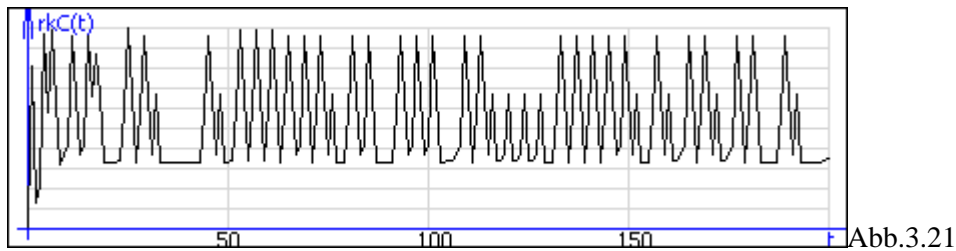


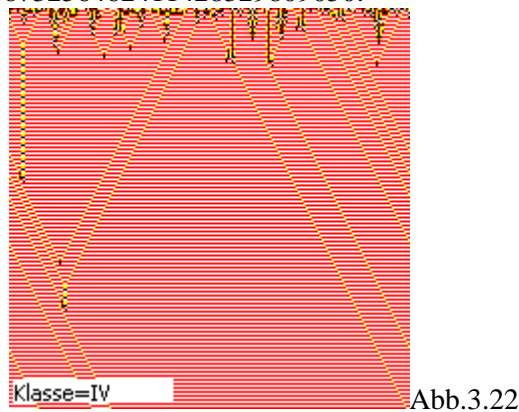
Abb.3.20

Erstaunlicherweise treten bei diesem Automaten immer wieder Zeitfenster der ungefähren Länge 50 auf, in denen nur wenige $rkC(t)$ -Werte angenommen wird, wie die nachfolgende Abbildung zeigt:

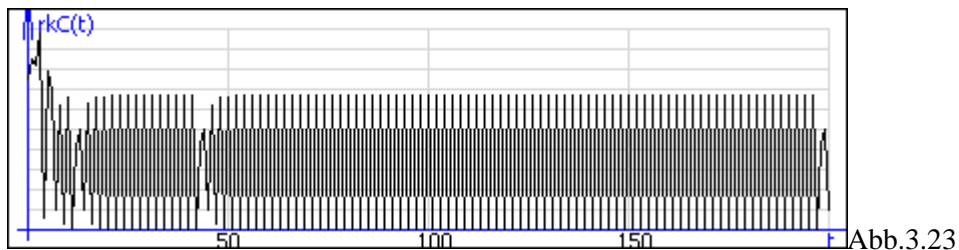


Deshalb stuft der Klassifikationsalgorithmus diesen Automaten in die Klasse IV ein.

esZA_1N4Z_203107557801526673256462411428329809050:

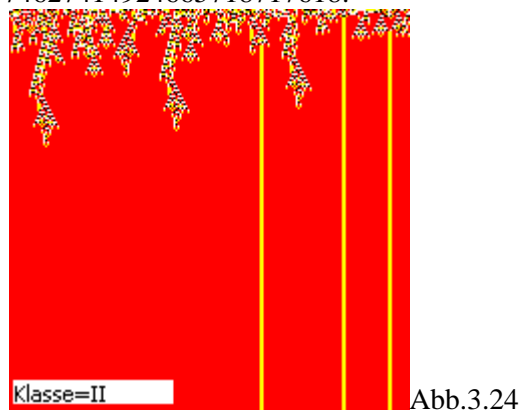


Offensichtlich ist die vom Algorithmus konstatierte Einordnung in die Klasse IV unangemessen. Es ist hier vielmehr die Zuordnung zur Klasse II gegeben. Auch hier liegt der Grund in der rkZ und in der Konzeption des Algorithmus, wie die nachfolgende Grafik verdeutlicht:



Nach ca. 30 - 50 Zeitschritten (aber meistens nicht eher!) gelangt die rkZ in einen periodischen Zustand kurzer Periodendauer oder weist genügend lange Zeitfenster mit periodischem Zustand auf, so dass dieser Automat als Automat der Klasse IV angesehen wird.

esZA_1N4Z_30310692902768374627414924665718717618:



Ein 1000-maliger Aufruf des Kernalgorithmus ergab hier 560 Mal die Klasse II und 440 Mal die Klasse IV, so dass der Hauptalgorithmus zur Einordnung des esZA in die Klasse II gelangt. Wie die Raum-Zeit-Entfaltung nahelegt, ist hier jedoch die subjektive Einstufung in die Klasse IV angemessen, vor allem, wenn man bedenkt, dass auch der klassische Klasse IV - Automat esZA_2N2Z_1771476584 keine wesentlich komplexere Raum-Zeit-Entfaltung aufweist (siehe Abb.3.25) als dieser Automat.



Klasse=IV

Abb.3.25 (esZA_2N2Z_1771476584)

Die Automatentypen esZA_1N5Z, esZA_1N6Z, esZA_1N7Z, esZA_1N8Z, esZA_1N9Z:

Stichproben mit jeweils 1000 Automaten ergaben in ca. 99,7% - 100% aller Fälle übereinstimmende subjektive und algorithmische Klassifikationen, wobei dieser Prozentsatz mit zunehmendem z anwächst. Dies hat folgenden Grund: Je größer bei festem n der Wert für z ist, desto größer ist der Anteil der Klasse III - Automaten des jeweiligen Automatentyps. Da der Algorithmus besonders gut die Klasse III - Automaten identifiziert, wird bei einer zufälligen Automatenstichprobe die Fehlerquote mit wachsendem z immer geringer. So gehörten beim Typ esZA_1N9Z alle 1000 Automaten der Stichprobe der Klasse III an und wurden auch als solche vom Algorithmus erkannt.

Die Automatentypen esZA_2N3Z, esZA_2N4Z, esZA_4N2Z:

Eine Stichprobe von 100 zufällig ausgewählten Automaten des Typs esZA_2N3Z ergab lediglich bei 2 Automaten Nichtübereinstimmung zwischen algorithmischer und subjektiver Klassifikation.

Beim Typ esZA_2N4Z ergab eine zufällig gewählte Stichprobe von 1000 Automaten, dass 997 Automaten der Klasse III angehörten, ein Automat der Klasse II und zwei Automaten der Klasse IV. Bei allen Automaten der Stichprobe gab es Übereinstimmung zwischen algorithmischer und subjektiver Klassifikation.

Beim Typ esZA_4N2Z ergab eine zufällig gewählte Stichprobe von 1000 Automaten, dass 982 Automaten der Klasse III angehörten, 14 Automaten der Klasse IV und 4 Automaten der Klasse II. Bei allen Automaten der Stichprobe gab es Übereinstimmung zwischen algorithmischer und subjektiver Klassifikation.

Immer wieder trifft man bei der Durchmusterung der Automaten eines gegebenen Automatentyps auf esZA, bei denen die subjektive Klassifikation nicht eindeutig ist. Ein guter Klassifikationsalgorithmus muss in solchen Fällen ebenfalls indifferente Klassifikationen vornehmen. Die folgenden Beispiele zeigen, dass der vorgeschlagene Klassifikationsalgorithmus dieses Kriterium recht ordentlich erfüllt.

So ist es schwierig, den Automaten esZA_1N3Z_2025356555774 subjektiv eindeutig einer der 4 Klassen zuzuordnen, da er sowohl Klasse III - als auch Klasse IV - Eigenschaften besitzt, wie die beiden folgenden Beispiele für Raum-Zeit-Entfaltungen zeigen. Nach der Abb.3.26 würde man den Automaten wegen seiner ausgeprägten Ordnungsstrukturen eher der Klasse IV zurechnen. Diese Ordnungsstrukturen sind in der Abb.3.27 kaum noch sichtbar, so dass hiernach die Einstufung in Klasse III sinnvoll erscheint. Ein menschlicher Betrachter studiert nun weitere Raum-Zeit-Entfaltungen, um zu einer Entscheidung zwischen Klasse III und Klasse IV zu gelangen.

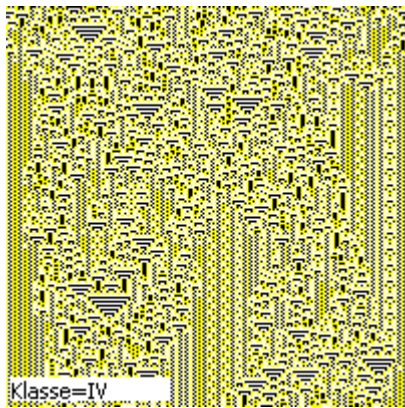


Abb.3.26

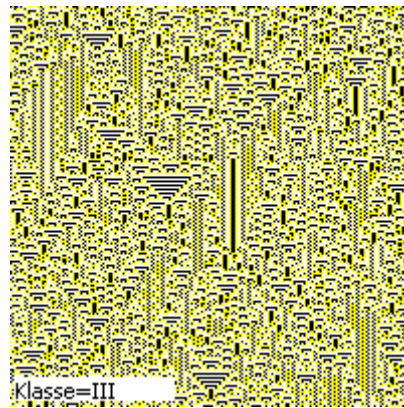


Abb.3.27

Auch der Klassifikationsalgorithmus legt seiner Einordnung mehrere Raum-Zeit-Entfaltungen zugrunde. Bei einem 40-maligen Aufruf des Kernalgorithmus (was ja dem subjektiven Studium 40 weiterer Raum-Zeit-Entfaltungen entspricht) kam der Algorithmus einmal zur Einordnung in die Klasse IV (siehe Abb.3.26) und ein anderes Mal zur Einordnung in die Klasse III (siehe Abb.3.27). Erst bei einem 1000-maligen Aufruf des Kernalgorithmus (was dem Studium von 1000 verschiedenen Raum-Zeit-Entfaltungen entspricht) kam der Algorithmus zu einem klareren Bild: Es wurde 434 Mal die Klasse III und 566 Mal die Klasse IV konstatiert, so dass der Hauptalgorithmus endgültig auf Klasse IV entschied. Bei diesem Beispiel korrespondieren also subjektive und algorithmische Klassifikation.

Die nachfolgenden zwei Grafiken zeigen zwei verschiedene Raum-Zeit-Entfaltungen des Automaten esZA_1N4Z_91822267272414081444151352839030631080:

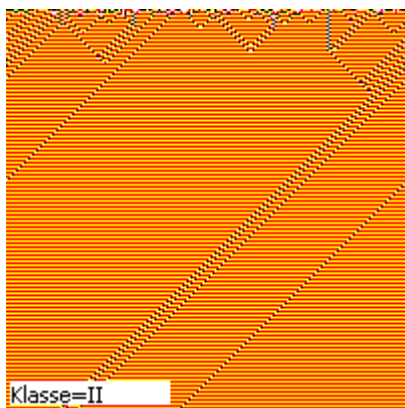


Abb.3.28

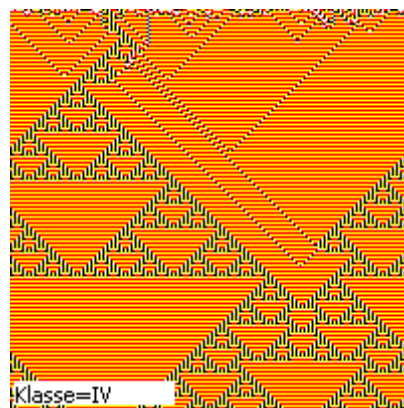


Abb.3.29

Je nach Startbelegung hat die Raum-Zeit-Entfaltung Klasse II-, bzw. Klasse IV-Eigenschaften, wie aus den beiden Abbildungen deutlich hervorgeht. Eine eindeutige subjektive Klassifikation scheint hier nicht möglich. Und wieder korrespondiert dazu die algorithmische Klassifikation mit synchronen Klassenzuweisungen, wie aus den obigen Grafiken ersichtlich wird. Die Einordnungen in die Klasse II (Abb.3.28) und in die Klasse IV (Abb.3.29) wurden nach jeweils 40 Aufrufen des Kernalgorithmus getroffen. Bei 1000-maligem Aufruf des Kernalgorithmus ergab sich ein deutlicheres Bild: 357 Mal wurde die Klasse II konstatiert, 20 Mal die Klasse III und 623 Mal die Klasse IV, so dass der Klassifikationsalgorithmus endgültig auf Klasse IV entschied. In der Tat: studiert man subjektiv mehrere Raum-Zeit-Entfaltungen dieses Automaten, so sind diejenigen mit Klasse IV - Merkmalen häufiger als die mit Klasse II - Eigenschaften, so dass auch ein menschlicher Betrachter geneigt ist, diesen Automaten in die Klasse IV einzuordnen. Bei diesem Beispiel korrespondieren also ebenfalls subjektive und algorithmische Klassifikation.

Als letztes Beispiel indifferenter Klassifikation soll der Automat esZA_2N2Z_3474639165 betrachtet werden:

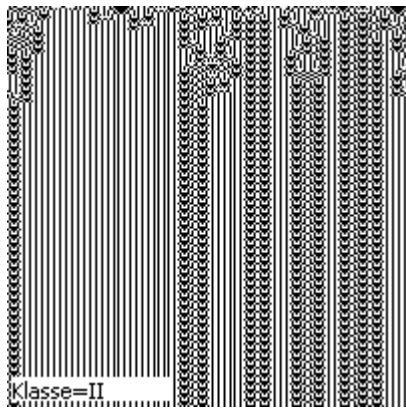


Abb.3.30

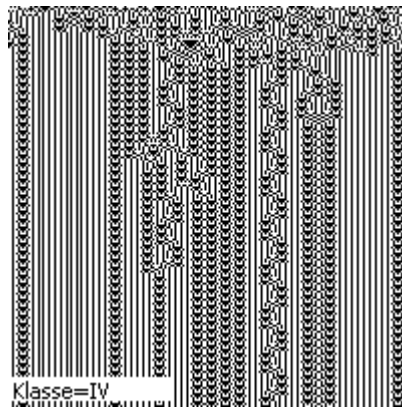


Abb.3.31

Nach Abb.3.30 würde man den Automaten in die Klasse II einstufen, wie es auch der Klassifikationsalgorithmus nach 40 Aufrufen des Kernalgorithmus durchgeführt hat. In der Abb.3.31 wird aber deutlich, dass der Automat auch Klasse IV - Eigenschaften aufweist, so dass eher die Einordnung in die Klasse IV gegeben scheint, so wie es in Abb.3.31 durch den Klassifikationsalgorithmus nach 40 Aufrufen des Kernalgorithmus bestätigt wird. 1000 Aufrufe des Kernalgorithmus ergaben 486 Mal Klasse II, 47 Mal Klasse III und 467 Mal Klasse IV, so dass der Hauptalgorithmus mit knapper Mehrheit auf Klasse II entschied. Ein subjektiver Betrachter ist nach dem Studium mehrerer Raum-Zeit-Entfaltungen auch eher geneigt, den Automaten der Klasse II zuzuordnen. Damit zeigt sich auch anhand dieses Beispiels eine akzeptable Korrespondenz algorithmischer und subjektiver Klassifikation.

Fazit:

Nimmt man die Ergebnisse der obigen Teiluntersuchungen zusammen und wagt eine Extrapolation, so lässt sich vermuten, dass bei ca. 99,7% der esZA eine Übereinstimmung zwischen algorithmischer und subjektiver Klassifikation vorliegt. Dies ist zwar nicht optimal, aber ein gutes Ergebnis. Die Übereinstimmung steigt rapide mit zunehmender Zustandsanzahl z an, da mit wachsendem z der Anteil der Klasse III - Automaten sehr hoch wird und Klasse III - Automaten vom Klassifikationsalgorithmus mit großer Sicherheit identifiziert werden. Außerdem ist immer wieder zu beobachten, dass die algorithmische Klassifikation immer dann indifferent ist, wenn auch die subjektive Klassifikation schwierig (wenn nicht gar unmöglich!) ist. Dies ist ein weiteres Zeichen für die Güte des Algorithmus. Die nachfolgenden Untersuchungen verwenden den beschriebenen Algorithmus. Die darauf beruhenden Ergebnisse werden mit einer Unsicherheit von ca. 0,3% akzeptiert.

4. Anwendungen des Klassifikationsalgorithmus

Der im 2. Abschnitt entwickelte und im 3. Abschnitt bezüglich seiner Güte überprüfte Klassifikationsalgorithmus für esZA wird nun selbst dazu verwendet, um zu Aussagen über esZA zu gelangen. Dabei wird angenommen, dass die bei Anwendung des Klassifikationsalgorithmus gewonnenen Klassenzuordnungen mit einer Unsicherheit von ca. 0,3% behaftet sind.

a) Klassenverteilung

So lässt sich z.B. untersuchen, welche prozentualen Anteile die vier Automatenklassen bei einem esZA gegebenen Typs aufweisen, d.h. welche sogenannte Klassenverteilung dieser Automat besitzt. Die folgende Tabelle Tab.4.1 zeigt die Klassenverteilung bei insgesamt 13 esZA-Typen. In der ersten Spalte sind die Anzahlen n der Nachbarzellen einer Zelle aufgelistet, in der zweiten Spalte die Anzahlen z der Zustände des Automaten. In der 3. Spalte findet man die Anzahlen der möglichen Codes beim Automaten des Typs esZA_nNzZ. In den letzten vier Spalten sind die relativen Häufigkeiten der vier Automatenklassen notiert. An den verschiedenen Tabellenfarben wird sichtbar, dass die Automaten zunächst nach der Nachbarschaftszahl n und dann nach der Zustandsanzahl z geordnet wurden.

Um zu den in der Tabelle angegebenen Häufigkeitsverteilungen zu gelangen, wurden jeweils 5000 Automaten des betreffenden Typs zufällig ausgewählt (außer beim Typ esZA_1N2Z, bei dem alle Automaten klassifiziert wurden), für die gewählten Automaten die algorithmische Klassifikation (Weltbreite=150; 50maliger Aufruf des Kernalgorithmus) berechnet und anschließend daraus die Klassenverteilung für diese 5000 Automaten ermittelt.

n	z	Anzahl der Automaten vom Typ esZA_nNzZ	Anteil Klasse I	Anteil Klasse II	Anteil Klasse III	Anteil Klasse IV
1	2	64	12,5%	62,5%	21,9%	3,1%
1	3	387 420 489	1,1%	46,3%	35,5%	17,1%
1	4	1 208 925 819 614 629 174 706 176 $\approx 1,20893 \cdot 10^{24}$	0,1%	16,9%	61,8%	21,2%
1	5	$\approx 2,64698 \cdot 10^{52}$	<0,02%	3,2%	86,4%	10,4%
1	6	$\approx 1,11444 \cdot 10^{98}$	<0,02%	0,3%	97,0%	2,7%
1	7	$\approx 4,35728 \cdot 10^{165}$	<0,02%	<0,02%	99,4%	0,6%
1	8	$\approx 1,23003 \cdot 10^{260}$	<0,02%	<0,02%	$\approx 100,0\%$	<0,02%
1	9	$\approx 2,93911 \cdot 10^{386}$	<0,02%	<0,02%	$\approx 100,0\%$	<0,02%
2	2	1 048 576	2,4%	45,2%	37,7%	14,7%
2	3	$\approx 2,57851 \cdot 10^{64}$	<0,02%	1,0%	94,8%	4,2%
2	4	$\approx 3,31616 \cdot 10^{327}$	<0,02%	<0,02%	$\approx 100,0\%$	<0,02%
3	2	$\approx 4,72237 \cdot 10^{21}$	0,1%	12,6%	73,1%	14,2%
4	2	$\approx 7,58855 \cdot 10^{81}$	<0,02%	0,7%	97,1%	2,2%

Tab.4.1

Die Tabelle offenbart einige Muster:

1. Besonders fällt der starke Zuwachs des Anteils der Klasse III - Automaten sowohl mit wachsendem n als auch mit wachsendem z ins Auge. Der Anteil dieser Automatenklasse scheint bei konstantem n mit wachsender Zahl z (und auch bei konstantem z mit wachsender Zahl n!) gegen 100% zu gehen.
2. Die Anteile der Automatenklassen I und II nehmen bei festem n mit wachsendem z und bei festem z mit wachsendem n ab. Es ist zu vermuten, dass sie jeweils gegen 0% streben.
3. Die Anteile der Klasse I - Automaten sind durchweg kleiner als die der Klasse II - Automaten.
4. Der Anteil der Klasse IV - Automaten steigt zunächst bei festem n und wachsendem z (aber auch bei festem z und wachsendem n!) an und sinkt dann nach Erreichen eines Maximums ab.
5. Das Absinken der relativen Häufigkeit der Klasse IV - Automaten mit wachsender Zahl z geschieht langsamer als bei den Klasse I - und Klasse II - Automaten, so dass der Klasse IV - Anteil ab einer gewissen Zahl z den Klasse I - Anteil und auch den Klasse II - Anteil übersteigt.
6. Der Automatentyp esZA_1N4Z besitzt mit ca. 21% den höchsten relativen Anteil an Klasse IV - Automaten.

Die nachfolgende Abbildung zeigt eine graphische Umsetzung eines Teils der obigen Tabelle, und zwar der Klassenverteilung der Automatentypen esZA_1NzZ mit z im Bereich von 2 bis 9:

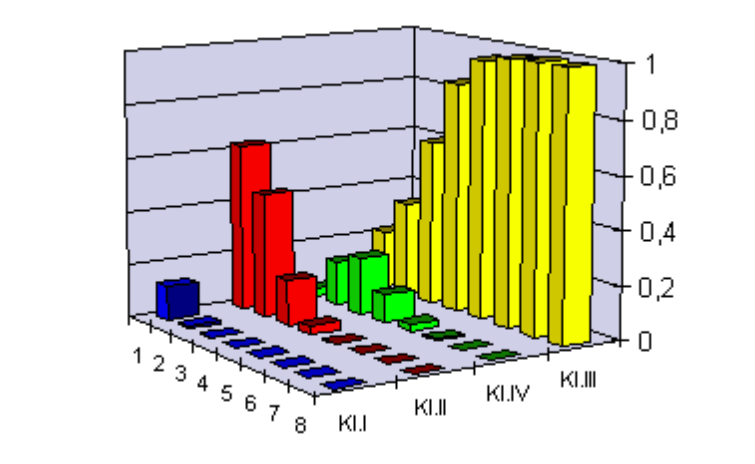


Abb.4.1

In einer weiteren Abbildung wird bei festem z=2 die Abhängigkeit der Klassenverteilung von n im Bereich von 1 bis 4 dargestellt:

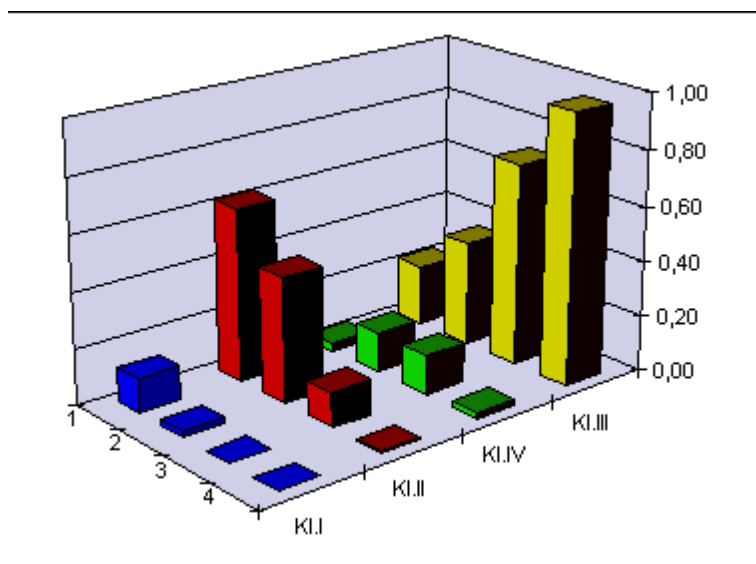
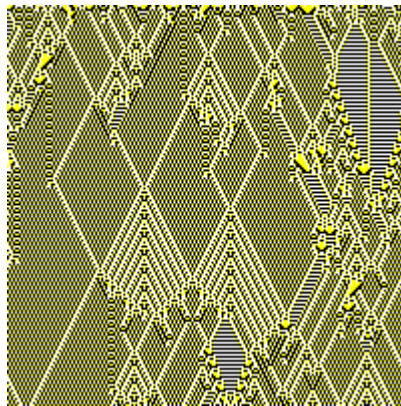


Abb.4.2

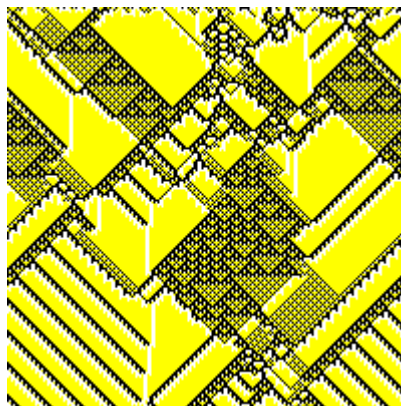
b)Automatisierte Suche nach Klasse IV - Automaten

Eine anderer Untersuchungsgegenstand bezieht sich auf Automaten der Klasse IV. Sie sind sicher die interessantesten unter den vier Automatenklassen und ihre Raum-Zeit-Entfaltungen sind ästhetisch häufig sehr ansprechend. Um besondere Eigenschaften dieser Automatenklasse aufzudecken, ist es sicherlich hilfreich, wenn man möglichst viele von ihnen kennt. Geht man nun der Reihe nach die Codes eines bestimmten Automatentyps durch, um Automaten der Klasse IV aufzufinden, so lässt sich in aller Regel (nach Kenntnisstand des Autors) anhand des Codes alleine nicht entscheiden, welcher Automatentyp gerade vorliegt. So wird die manuelle Suche nach Automaten der Klasse IV u.U. sehr zeitaufwändig, insbesondere wenn man bedenkt, dass mit größer werdender Nachbarschaftszahl n und Zustandsanzahl z die Automaten der Klasse IV immer seltener auftreten (siehe Tab.4.1). Hier leistet der Klassifikationsalgorithmus gute Dienste: Man lässt ihn der Reihe nach (oder nach dem Zufallsprinzip) die Automaten eines bestimmten Typs esZA_nNzZ klassifizieren und markiert diejenigen Automaten, die der Klassifikationsalgorithmus als Klasse IV - Automaten konstatiert. Da die Unsicherheit mit ca. 0,3% recht klein ist (siehe oben), findet man so mit hoher Wahrscheinlichkeit nach ausreichender Wartezeit hinreichend viele Automaten der Klasse IV.

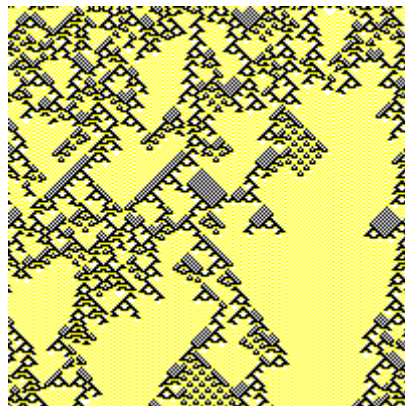
Die nachfolgenden 11 Abbildungen zeigen Beispiele einiger auf diese Weise ermittelter Klasse IV - Automaten, deren Raum-Zeit-Entfaltungen nach Meinung des Autors ästhetisch ansprechend sind:



esZA_N3Z_2047127316893: Abb.4.3



esZA_1N3Z_2516580825751: Abb.4.4



esZA_1N3Z_3221433703281: Abb.4.5

esZA_1N4Z_109376237795525598685152182496903313608:

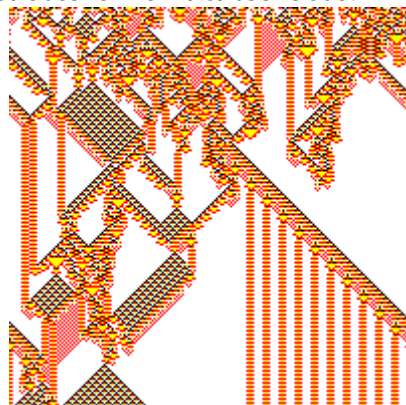


Abb.4.6

esZA_1N4Z_201546222498561028214313401087722311199:

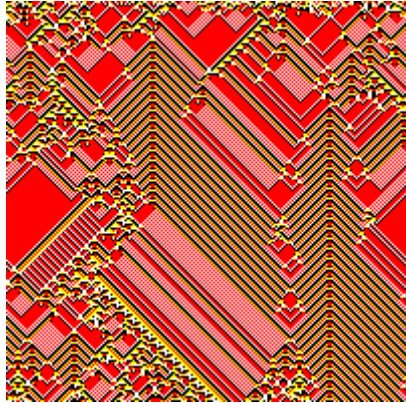


Abb.4.7

esZA_1N4Z_254533936725584910422158283895346876867:

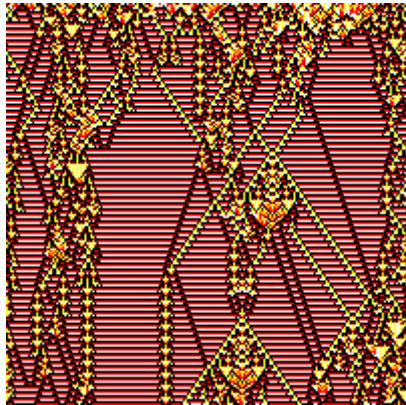
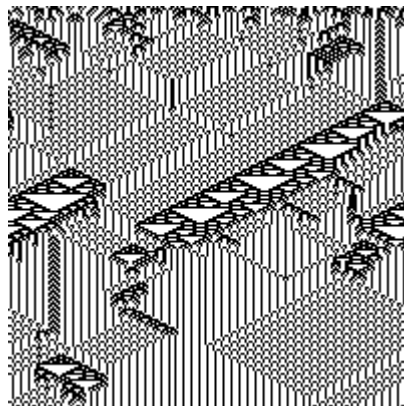
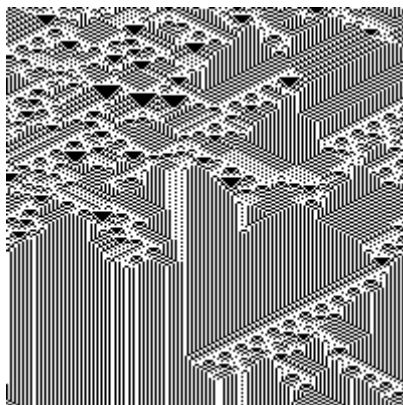


Abb.4.8



esZA_2N2Z_2107743450:

Abb.4.9



esZA_2N2Z_3476719657:

Abb.4.10

esZA_2N2Z_3607678089:

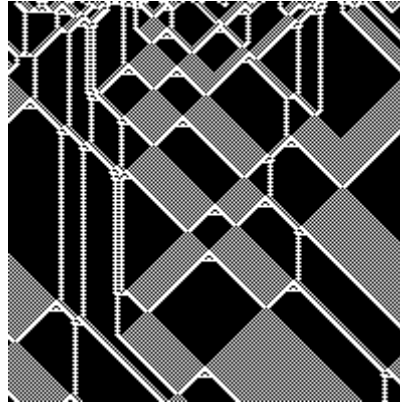


Abb.4.11

esZA_2N3Z_8610101844108993690149127860183537212688064694976412104285910277369271209143
9773043952501672248340502916745135354145:

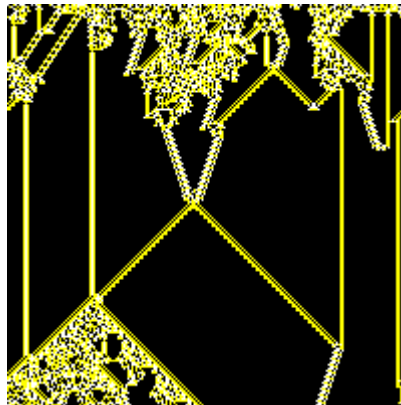


Abb.4.12

esZA_3N2Z_229658604473554615225339356748274879560:

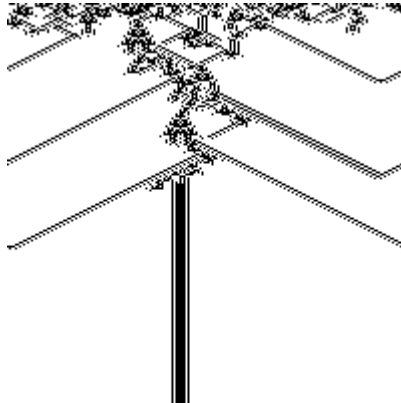


Abb.4.13

c) Häufungen von Klasse IV - Automaten

Eine weitere Fragestellung betrifft ebenfalls die interessanten Klasse IV - Automaten:

Bei genügend großer Nachbarschaftszahl und/oder Zustandszahl ist die Anzahl der Codes eines bestimmten Automatentyps enorm groß (siehe z.B. Tab.4.1). Man kann in solchen Fällen regelrecht von einem *Zahlenraum der Codes* sprechen, in dem hin und wieder Automaten der Klasse IV auftreten. Ihr Auftreten wird mit wachsendem n und z immer seltener. Die Frage ist nun: Gibt es Häufungen von Klasse IV - Automaten in diesem Zahlenraum, d.h. treten diese Automaten im Zahlenraum der Codes vorwiegend isoliert auf oder kommt es vor, dass Klasse IV - Automaten mit Automaten benachbart sind, die ebenfalls der Klasse IV angehören?

Um diese Frage anzugehen, muss man zunächst festlegen, wie der Begriff „benachbarte Codes“ verstanden werden soll. Sicherlich ist es im Hinblick auf die Problemstellung nicht angemessen, zwei Codes als benachbart anzusehen, wenn ihre Dezimaldarstellungen aufeinanderfolgen. Dazu ein einfaches Beispiel, das den Sachverhalt verdeutlicht: Die Automaten esZA_1N2Z_127 und esZA_1N2Z_128 sind bezüglich ihrer Dezimaldarstellung unmittelbar benachbart, weisen aber ein völlig verschiedenes Raum-Zeit-Verhalten auf und gehören verschiedenen Klassen an, wie die beiden folgenden Grafiken zeigen. In der Abb.4.13 ist eine Raum-Zeit-Entfaltung des Klasse II - Automaten esZA_1N2Z_127 dargestellt und in der Abb.4.14 eine Raum-Zeit-Entfaltung des Klasse I - Automaten esZA_1N2Z_128:

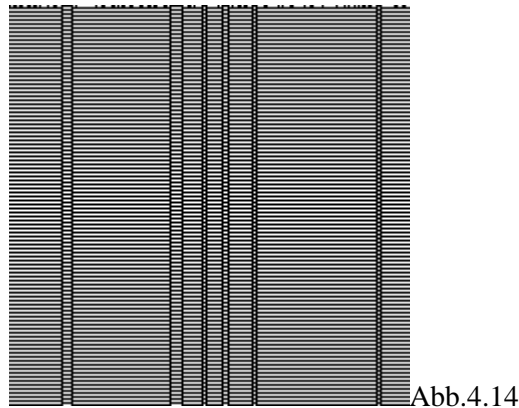


Abb.4.15

Der Grund wird sofort klar, wenn man sich die Regelsätze der beiden Automaten ansieht.

Der Regelsatz von esZA_1N2Z_127 ist:

$$f([0,0,0])=1; f([0,0,1])=1; f([0,1,0])=1; f([0,1,1])=1; f([1,0,0])=1; f([1,0,1])=1; f([1,1,0])=1; f([1,1,1])=0 .$$

Der Regelsatz von esZA_1N2Z_128 ist:

$$f([0,0,0])=0; f([0,0,1])=0; f([0,1,0])=0; f([0,1,1])=0; f([1,0,0])=0; f([1,0,1])=0; f([1,1,0])=0; f([1,1,1])=1 .$$

Die Regelsätze weichen stark voneinander ab, und daraus resultieren natürlich die unterschiedlichen Raum-Zeit-Entfaltungen und somit auch die verschiedenen Klassenzugehörigkeiten. Die beiden Automaten sind zwar bezüglich ihrer Dezimalcodes benachbart, nicht aber bezüglich ihrer Regelsätze.

Ein zweites Beispiel:

Die Automaten esZA_1N2Z_72 und esZA_1N2Z_76 sind bezüglich ihrer Codes in Dezimaldarstellung nicht benachbart. Schaut man sich aber die Regelsätze der beiden Automaten an, so erkennt man sofort, dass sie bezüglich der Regelsätze sehr wohl benachbart sind.

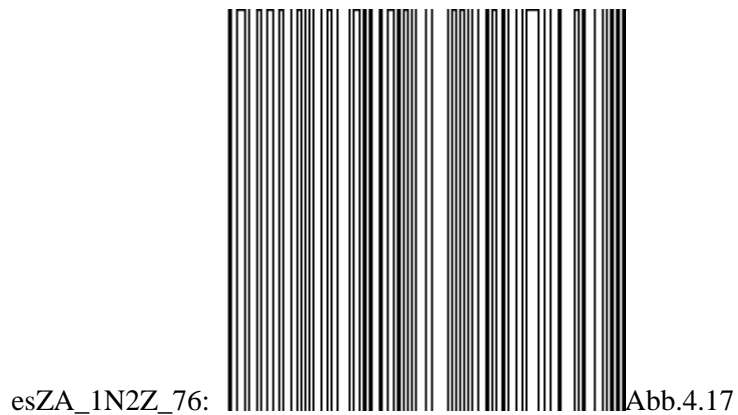
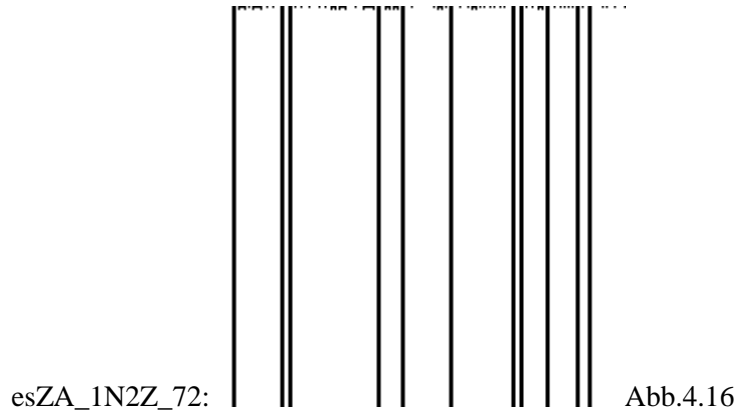
Der Regelsatz von esZA_1N2Z_72 ist:

$$f([0,0,0])=0; f([0,0,1])=0; f([0,1,0])=0; f([0,1,1])=1; f([1,0,0])=0; f([1,0,1])=0; f([1,1,0])=1; f([1,1,1])=0 .$$

Der Regelsatz von esZA_1N2Z_76 ist:

$f([0,0,0])=0$; $f([0,0,1])=0$; $f([0,1,0])=1$; $f([0,1,1])=1$; $f([1,0,0])=0$; $f([1,0,1])=0$; $f([1,1,0])=1$; $f([1,1,1])=0$.

Die Regelsätze unterscheiden sich lediglich in $f([0,1,0])$. Ansonsten sind sie identisch. In der Tat gehören beide Automaten der Klasse II an, wie die beiden folgenden Abbildungen belegen:



Dass nicht immer, wenn die Regelsätze zweier Automaten sehr ähnlich sind, auch die Entfaltungen ähnlich sein müssen, ist allerdings auch klar! Dazu ebenfalls ein Beispiel:

Die Automaten esZA_1N2Z_90 und esZA_1N2Z_91 sind sowohl bezüglich ihrer Dezimalcodes benachbart als auch sehr ähnlich bezüglich ihrer Regelsätze.

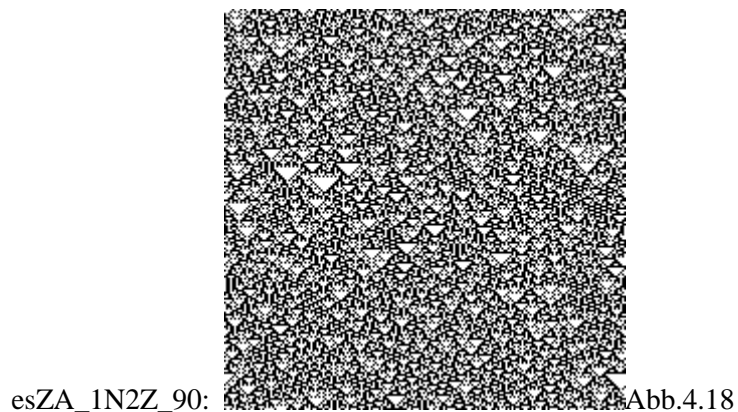
Der Regelsatz von esZA_1N2Z_90 lautet:

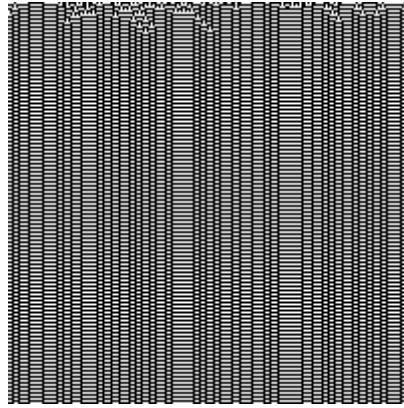
$f([0,0,0])=0$; $f([0,0,1])=1$; $f([0,1,0])=0$; $f([0,1,1])=1$; $f([1,0,0])=1$; $f([1,0,1])=0$; $f([1,1,0])=1$; $f([1,1,1])=0$.

Der Regelsatz von esZA_1N2Z_91 lautet:

$f([0,0,0])=1$; $f([0,0,1])=1$; $f([0,1,0])=0$; $f([0,1,1])=1$; $f([1,0,0])=1$; $f([1,0,1])=0$; $f([1,1,0])=1$; $f([1,1,1])=0$.

Trotzdem gehört esZA_1N2Z_90 der Klasse III an und esZA_1N2Z_91 der Klasse II:





esZA_1N2Z_91: Abb.4.19

Nichtsdestoweniger scheint es sinnvoller zu sein, Nachbarschaft zweier Automaten über Ähnlichkeit der Regelsätze zu definieren als über Nähe ihrer Dezimalcodes, da die Klassenzugehörigkeit und das Raum-Zeit-Verhalten durch den Regelsatz festgelegt wird, und der Dezimalcode lediglich eine Verschlüsselung des Regelsatzes bedeutet, um die Automaten eines Typs übersichtlich und ökonomisch zu erfassen.

Wie in den Grundlagen im 1. Abschnitt dargelegt, handelt es sich beim Regelsatz eines ZA um eine Funktion f von der Menge der $2n+1$ -Tupel der möglichen Nachbarschaftszustandskonfigurationen in die Zustandsmenge $Z = \{0, 1, \dots, z-1\}$. Zwei verschiedene Regelsätze unterscheiden sich in dem Funktionswert mindestens eines $2n+1$ -Tupels voneinander. Je größer die Anzahl der übereinstimmenden Wertepaare der beiden Regelsätze sind, desto weniger unterscheiden sich die Regelsätze voneinander und desto 'benachbarter' sind sie somit. Dies soll nun im Hinblick auf symmetrische Regelsätze näher präzisiert werden:

Schaut man sich Menge der $2n+1$ -Tupel der möglichen Nachbarschaftszustandskonfigurationen an, so gibt es zum Einen die zur mittleren Koordinate symmetrischen $2n+1$ -Tupel und zum Anderen die zur mittleren Koordinate asymmetrischen $2n+1$ -Tupel. Zu jedem asymmetrischen $2n+1$ -Tupel existiert genau ein zweites dazu verschiedenes sog. korrespondierendes $2n+1$ -Tupel, das aus dem ersten durch Spiegelung an der mittleren Koordinate hervorgeht. Ein Beispiel: Ist $n=3$ und $z=4$, so ist beispielsweise $[1,0,2,3,2,0,1]$ ein solches symmetrisches 7-Tupel. Das dazu korrespondierende 7-Tupel ist das 7-Tupel selbst! Das 7-Tupel $[0,2,3,1,0,1,2]$ dagegen ist ein asymmetrisches 7-Tupel. Das dazu korrespondierende 7-Tupel ist $[2,1,0,1,3,2,0]$. Eindimensionale symmetrische ZA, um die es in dieser Arbeit geht, haben die definierende Eigenschaft, dass die Funktionswerte von f für ein $2n+1$ -Tupel und das dazu korrespondierende $2n+1$ -Tupel gleich sind.

Geht man nun von einem symmetrischen Regelsatz aus und ändert den Funktionswert eines einzigen $2n+1$ -Tupels von z_1 auf z_2 , so entsteht ein neuer Regelsatz. Ist das betrachtete $2n+1$ -Tupel selbst symmetrisch, so ist auch der neue Regelsatz symmetrisch. Ist das betrachtete $2n+1$ -Tupel jedoch asymmetrisch, so bleibt der neue Regelsatz genau dann symmetrisch, wenn man auch den Funktionswert des dazu korrespondierenden $2n+1$ -Tupels auf z_2 setzt. Dies bedeutet, dass man bei einem symmetrischen Regelsatz entweder *einen* Funktionswert (bei einem symmetrischen $2n+1$ -Tupel) oder sofort *zwei* Funktionswerte (bei einem asymmetrischen $2n+1$ -Tupel) abändern muss, um zu einem neuen symmetrischen Regelsatz zu kommen, der eine minimale Anzahl abweichender Funktionswerte aufweist.

In diesem Sinne soll folgende Definition getroffen werden:

Zwei symmetrische Regelsätze heißen unmittelbar benachbart, wenn sie in dem Funktionswert genau eines $2n+1$ -Tupels nicht übereinstimmen, falls das zugehörige $2n+1$ -Tupel symmetrisch ist, oder in den Funktionswerten von genau zwei $2n+1$ -Tupeln nicht übereinstimmen, falls die beiden $2n+1$ -Tupel zueinander korrespondieren.

Anmerkung: Es folgt aus der Bedingung der Symmetrie der beiden Regelsätze, dass im zweiten Fall die beiden Funktionswerte gleich sein müssen.

Zwei esZA sollen unmittelbar benachbart heißen, falls ihre Regelsätze unmittelbar benachbart sind.

Drei Beispiele sollen die Definition veranschaulichen:

1. Beispiel: Es gelte $n=1$ und $z=2$.

Der Regelsatz eines Automaten vom Typ esZA_1N2Z laute:

$f([0,0,0])=0$; $f([0,0,1])=1$; $f([0,1,0])=0$; $f([0,1,1])=1$; $f([1,0,0])=1$; $f([1,0,1])=0$; $f([1,1,0])=1$; $f([1,1,1])=0$.

Der Regelsatz eines zweiten Automaten vom Typ esZA_1N2Z laute:

$g([0,0,0])=0$; $g([0,0,1])=1$; $g([0,1,0])=0$; $g([0,1,1])=1$; $g([1,0,0])=1$; $g([1,0,1])=1$; $g([1,1,0])=1$; $g([1,1,1])=0$.

Die Regelsätze sind unmittelbar benachbart, da sie lediglich bezüglich des symmetrischen 3-Tupels $[1,0,1]$ voneinander abweichen.

2. Beispiel: Es gelte $n=1$ und $z=2$.

Der Regelsatz eines Automaten vom Typ esZA_1N2Z laute:

$f([0,0,0])=0$; $f([0,0,1])=1$; $f([0,1,0])=0$; $f([0,1,1])=1$; $f([1,0,0])=1$; $f([1,0,1])=0$; $f([1,1,0])=1$; $f([1,1,1])=0$.

Der Regelsatz eines zweiten Automaten vom Typ esZA_1N2Z laute:

$g([0,0,0])=0$; $g([0,0,1])=0$; $g([0,1,0])=0$; $g([0,1,1])=1$; $g([1,0,0])=0$; $g([1,0,1])=0$; $g([1,1,0])=1$; $g([1,1,1])=0$.

Die Regelsätze sind unmittelbar benachbart, da sie lediglich bezüglich des 3-Tupels $[1,0,0]$ und des dazu korrespondierenden 3-Tupels $[0,0,1]$ voneinander abweichen.

3. Beispiel: Es gelte $n=1$ und $z=2$.

Der Regelsatz eines Automaten vom Typ esZA_1N2Z laute:

$f([0,0,0])=0$; $f([0,0,1])=1$; $f([0,1,0])=0$; $f([0,1,1])=1$; $f([1,0,0])=1$; $f([1,0,1])=0$; $f([1,1,0])=1$; $f([1,1,1])=0$.

Der Regelsatz eines zweiten Automaten vom Typ esZA_1N2Z laute:

$g([0,0,0])=1$; $g([0,0,1])=1$; $g([0,1,0])=0$; $g([0,1,1])=1$; $g([1,0,0])=1$; $g([1,0,1])=1$; $g([1,1,0])=1$; $g([1,1,1])=0$.

Die Regelsätze sind nicht unmittelbar benachbart, da sie bezüglich des symmetrischen 3-Tupels $[1,0,1]$ und des symmetrischen 3-Tupels $[0,0,0]$ voneinander abweichen.

Die Anzahl n_{ubR} der zu einem Regelsatz unmittelbar benachbarten Regelsätze hängt natürlich von n und z ab. Diese Anzahl n_{ubR} lässt sich berechnen:

Zu gegebenem n und z existieren z^{2n+1} verschiedene $2n+1$ -Tupel, d.h. Nachbarschaftszustandskonfigurationen. Davon sind z^{n+1} symmetrisch bezüglich ihrer mittleren Koordinate. Also existieren insgesamt $z^{2n+1} - z^{n+1}$ zu ihrer mittleren Koordinate asymmetrische $2n+1$ -Tupel. Ist nun ein Regelsatz vorgegeben, so gibt es zunächst einmal $(z-1) \cdot z^{n+1}$ Möglichkeiten, zu einem unmittelbar benachbarten Regelsatz zu gelangen, indem man den Funktionswert eines einzigen symmetrischen $2n+1$ -Tupels abändert. Außerdem kann man zu einem unmittelbar benachbarten Regelsatz gelangen, indem man den Funktionswert eines asymmetrischen und seines korrespondierenden $2n+1$ -Tupels in gleicher Weise abwandelt. Dafür gibt es aber, da man ja immer die Funktionswerte zweier $2n+1$ -Tupel gleichzeitig in derselben Weise variiert, $(z-1) \cdot \frac{1}{2} \cdot (z^{2n+1} - z^{n+1})$ Möglichkeiten. Damit existieren zu einem symmetrischen Regelsatz insgesamt

$$(z-1) \cdot \frac{1}{2} \cdot (z^{2n+1} - z^{n+1}) + (z-1) \cdot z^{n+1} = \frac{1}{2} \cdot (z-1) \cdot z^{n+1} \cdot (z^n + 1)$$

unmittelbar benachbarte Regelsätze.

$$n_{\text{ubR}} = \frac{1}{2} \cdot (z-1) \cdot z^{n+1} \cdot (z^n + 1)$$

Die folgende Tabelle zeigt für einige Automatentypen die Anzahl der zu einem Regelsatz des Typs esZA_nNzZ unmittelbar benachbarten Regelsätze gleichen Typs:

n	z	n _{ubR}
1	2	6
1	3	36
1	4	120
1	5	300
1	6	630
1	7	1176
1	8	2016
1	9	3240
2	2	20
2	3	270
2	4	1632
3	2	72
4	2	272

Tab.4.2

Um herauszufinden, wie häufig Klasse IV - Automaten von anderen Klasse IV - Automaten unmittelbar benachbart werden, geht man folgendermaßen vor:

Man sucht zunächst mit Hilfe eines Zufallszahlengenerators und des Klassifikationsalgorithmus nach einem Klasse IV - Automaten. Dann ermittelt man (wieder mit dem Klassifikationsalgorithmus) die Anzahl $n_{ubR,IV}$ aller zu diesem Automaten unmittelbar benachbarten Automaten der Klasse IV. Der Quotient $n_{ubR,IV}/n_{ubR}$, der mit $n_{ubR,IV,rel}$ bezeichnet werden soll, gibt dann die relative Häufigkeit der Automaten aus der Nachbarschaft an, die der Klasse IV angehören.

Auch hier soll ein Beispiel den Sachverhalt aufhellen:

Der Zufallszahlengenerator und der Klassifikationsalgorithmus lieferten den Automaten esZA_2N2Z_2748081673 als Automaten der Klasse IV, was durch die subjektive Klassifikation bestätigt wird:

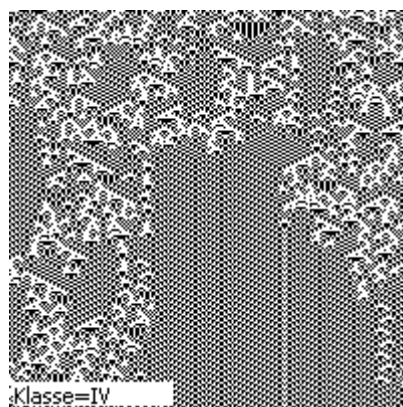


Abb.4.20

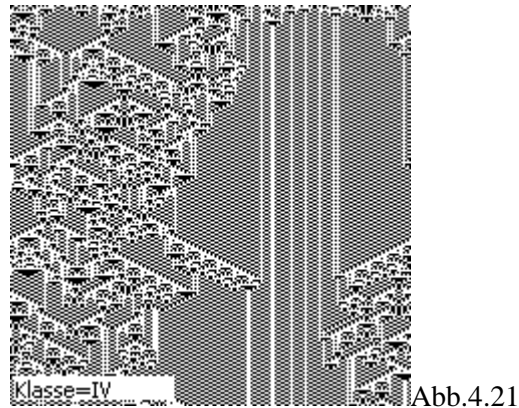
Die folgenden 20 Automaten (siehe Tab.4.2) sind diesem esZA unmittelbar benachbart:

esZA_2N2Z_2748081672, esZA_2N2Z_2748147211, esZA_2N2Z_2748081933, esZA_2N2Z_2731304449, esZA_2N2Z_2748081689, esZA_2N2Z_2749130281, esZA_2N2Z_2748085833, esZA_2N2Z_3016517257, esZA_2N2Z_2747819017, esZA_2N2Z_2748080649, esZA_2N2Z_2815192585, esZA_2N2Z_2743879177, esZA_2N2Z_2748065289, esZA_2N2Z_3821856265, esZA_2N2Z_2748212745, esZA_2N2Z_2714002953, esZA_2N2Z_2750178825, esZA_2N2Z_2202822153, esZA_2N2Z_2882299401, esZA_2N2Z_600598025 .

Der Klassifikationsalgorithmus identifiziert nun von diesen 20 Automaten korrekt alle 10 esZA, die der Klasse IV angehören:

esZA_2N2Z_2748081689, esZA_2N2Z_2748085833, esZA_2N2Z_3016517257, esZA_2N2Z_2748080649, esZA_2N2Z_2815192585, esZA_2N2Z_2743879177, esZA_2N2Z_2748065289, esZA_2N2Z_3821856265, esZA_2N2Z_2202822153, esZA_2N2Z_600598025 .

Die nachfolgende Abbildung zeigt exemplarisch die Raum-Zeit-Entfaltung eines dieser unmittelbar benachbarten Klasse IV - Automaten (esZA_2N2Z_2748080649):



Für den Automaten esZA_2N2Z_2748081673 hat der Quotient $n_{ubR,IV,rel} = n_{ubR,IV}/n_{ubR}$ somit den Wert 10/20, also 50%.

Wiederholt man bei einem gegebenen Automatentyp die beschriebene Untersuchung genügend oft und bildet man anschließend den Mittelwert über all diese Quotienten $n_{ubR,IV,rel} = n_{ubR,IV}/n_{ubR}$, so hat man ein ungefähres Maß dafür, wie häufig Klasse IV - Automaten im Regelsatzraum von anderen Klasse IV - Automaten umgeben sind. Der Mittelwert über die vom Algorithmus gebildeten $n_{ubR,IV,rel}$ -Werte soll *mittlere Klasse IV - Nachbarschaftsdichte* (abgekürzt: m_{IV_ND}) genannt werden. Die folgende Tabelle zeigt einige Ergebnisse:

n	z	n_{ubR}	m_{IV_ND}
1	2	6	0%
1	3	36	44,3%
1	4	120	66,3%
1	5	300	79,7%
1	6	630	87,1%
1	7	1176	87,9%
1	8	2016	95,0%
1	9	3240	77,4%
2	2	20	30,0%
2	3	270	90,2%
2	4	1632	80,35%
3	2	72	65,4%
4	2	272	88,7%

Tab.4.3

Die Zahlen sind bemerkenswert: obwohl die Klasse IV - Anteile bei fester Nachbarschaftszahl n mit zunehmender Zustandsanzahl z sinken (siehe Tab.4.2), steigt die mittlere Klasse IV - Nachbarschaftsdichte r_{IV_ND} an! Dies bedeutet nichts anderes, als dass die Klasse IV - Automaten mit wachsendem z zwar immer seltener werden, deren lokale relative Häufigkeit jedoch anwächst, und zwar auf beträchtliche Werte!

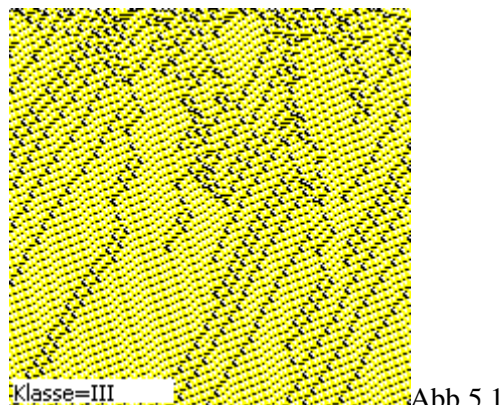
Die Wahrscheinlichkeit, dass ein unmittelbar benachbarter esZA ebenfalls der Klasse IV angehört, beträgt bei einigen Automatentypen mehr als 90%. Umgangssprachlich ausgedrückt heißt es, dass die Klasse IV - Automaten zur 'Haufenbildung' bzw. zur 'Klumpenbildung' neigen: wo einer ist, da sind mit großer Wahrscheinlichkeit auch noch mehrere. Und noch weitergehend: je größer z ist, desto stärker ist die 'Klumpigkeit'. Die globale Dichte der Klasse IV - Automaten nimmt ab, und deshalb müssen die lokalen Dichten mit wachsender Zustandsanzahl z immer größere Unterschiede aufweisen. Dies erinnert ein wenig an die ungleichmäßige Verteilung der sichtbaren Materie im Universum: riesengroße Bereiche mit äußerst geringer Materiedichte (z.B. interstellarer Raum zwischen Galaxien) versus kleine Raumregionen mit relativ großer Materiekonzentration (z.B. Planetensysteme oder große Sterne).

Die Zahlenwerte aus Tab.4.3 sind allerdings nur mit Vorsicht zu genießen, da sie auf der Auswertung von nur recht wenigen Klasse IV - Automaten beruhen: Wie aus den vorangegangenen Untersuchungen hervorgeht, werden mit zunehmendem z die Klasse IV - Automaten extrem selten. Der Algorithmus zur Ermittlung der mittleren Klasse IV - Nachbarschaftsdichte muss also u.U. sehr lange suchen, bis überhaupt ein Klasse IV - Automat gefunden wird. Außerdem kommt hinzu, dass bei größerem n und z die Anzahl der unmittelbar benachbarten esZA schnell sehr groß wird. Dies verlängert auch noch einmal die Rechenzeiten extrem, da u.U. pro aufgefundenem Klasse IV - Automat Tausende unmittelbar benachbarter Automaten klassifiziert werden müssen. So benötigte ein 600 MHz - Pentium - Prozessor für einige Tabellenwerte in Tab.4.3 über 24 Stunden Rechenzeit. Damit überhaupt in vertretbarem Zeitaufwand einigermaßen vernünftige Abschätzungen gegeben werden konnten, wurden pro Tabellenwert aus Tab.4.3 nur ca. 20 durch den Klassifikationsalgorithmus ermittelte Klasse IV - Automaten zugrundegelegt. Diese schmale Datenbasis ist natürlich völlig unzureichend und muss bei einer genaueren Untersuchung verbreitert werden. Trotzdem werfen die Tabellenwerte ein deutliches Schlaglicht auf die Entwicklung der mittleren Klasse IV - Nachbarschaftsdichten.

5.Schlussbetrachtung

Der in dieser Arbeit entwickelte und diskutierte Klassifikationsalgorithmus arbeitet mit recht hoher Sicherheit bei eindimensionalen zellulären Automaten, die einen symmetrischen Regelsatz besitzen. Er lässt sich mit eingeschränkter Sicherheit auch auf asymmetrische eZA anwenden. So wird beispielsweise der Automat eZA_1N2Z_110 (siehe z.B. Abb.1.3 und Abb.1.4), der auch in Stephen Wolframs Arbeiten ausführlich diskutiert wird (siehe [Wo2], S.675 - 693), vom Klassifikationsalgorithmus eindeutig als Klasse IV - Automat erkannt. Leider existieren jedoch etliche asymmetrische eZA, bei denen der Algorithmus eine unangemessene Klassifikation angibt. Eine erste Analyse zeigt den Grund auf:

So gehört beispielsweise der asymmetrische Automat eZA_1N3Z_4004769746578 der Klasse IV an. Trotzdem weist der Klassifikationsalgorithmus ihn deutlich als Klasse III - Automaten aus, wie die folgende Abb.5.1 zeigt:



Der Grund für die unangemessene Klassifikation wird erkennbar, wenn man sich die rkZ des Automaten anschaut. Die folgenden drei Abbildungen zeigen die ersten 600 Generationen der rkZ dieses Automaten, ausgehend von einer zufälligen Startbelegung:

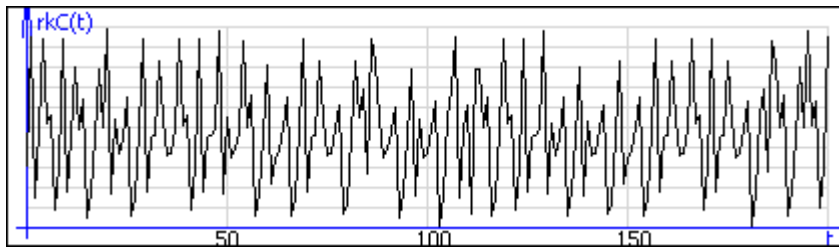


Abb.5.2

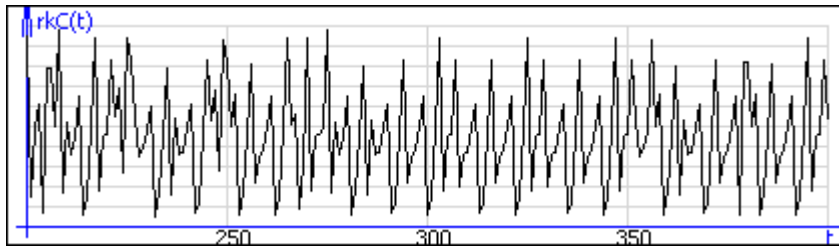


Abb.5.3

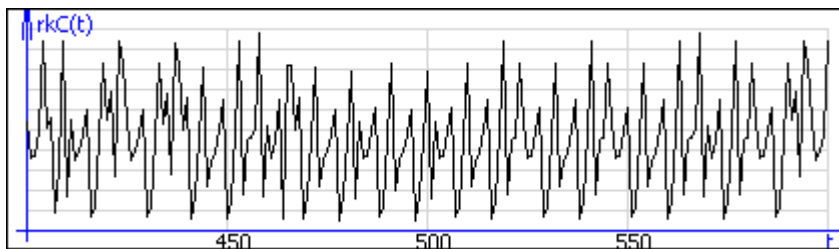


Abb.5.4

Weder wird die rkZ nach überschaubar vielen Zeitschritten periodisch mit kleiner Periodendauer noch gibt es Zeitfenster genügender Länge, in denen nur sehr wenige rkC(t)-Werte angenommen werden. Deshalb entscheidet der Klassifikationsalgorithmus hier aufgrund seines Design auf Klasse III. Im Gegensatz dazu ist die subjektive Klassifikation bei diesem Automaten eindeutig Klasse IV: Es finden sich immer wieder ausgedehnte Raumbereiche, die sehr geordnet sind, und diese Raumbereiche bleiben über begrenzte Zeiträume erhalten. Diese Raumbereiche haben jedoch die besondere Eigenschaft, dass ihre linken und rechten Begrenzungen gleichzeitig nach links bzw. rechts durch den Zellraum wandern, so dass sie sich in der rkZ nicht (oder zumindest nicht häufig genug!) als konstante Phasen bzw. Phasen mit nur wenig angenommenen rkC(t)-Werten widerspiegeln.

Dieses Phänomen gibt es bei immer wieder bei eZA mit asymmetrischem Regelsatz, und immer dann versagt der Klassifikationsalgorithmus. So z.B. auch beim Klasse IV - Automaten eZA_1N3Z_7367854177583, wie die nachfolgende Grafik zeigt:

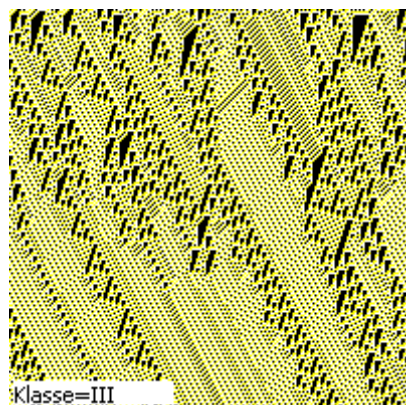


Abb.5.5

Bei esZA treten Automaten mit einer Raum-Zeit-Entfaltung vergleichbar mit der der Automaten eZA_1N3Z_4004769746578 und eZA_1N3Z_7367854177583 nicht auf, so dass der Klassifikationsalgorithmus bei ihnen wegen dieser Schwierigkeit nicht versagt. Die beim Automaten eZA_1N3Z_4004769746578 aufgedeckte Schwäche des Klassifikationsalgorithmus ist nicht geringfügig, sondern zentral! Die rkZ scheint prinzipiell ungeeignet, Klasse IV -Automaten der Art

eZA_1N3Z_4004769746578 von Klasse III - Automaten abzugrenzen. Der Verfasser sieht derzeit keinen Weg, den Klassifikationsalgorithmus unter Einbeziehung der rkZ geeignet zu 'reparieren'. Somit stellt sich das folgende zentrale

1.Problem:

Wie lässt sich der Klassifikationsalgorithmus modifizieren bzw. neu konzipieren, damit er sich auf alle eZA anwenden lässt?

Weitere Probleme, die sich im Zuge der durchgeführten Untersuchungen herausstellten, sind die folgenden:

2.Problem:

Wie lässt sich der Klassifikationsalgorithmus modifizieren, damit seine Unsicherheit, die derzeit bei maximal ca. 0,3% liegt, bei esZA geringer wird?

3.Problem:

Gibt es einen Grenzwert unter 100%, gegen den die mittleren Klasse IV - Nachbarschaftsdichten streben? Falls ja, wie groß ist dieser Wert?

4.Problem:

Die Ergebnisse zu den mittleren Klasse IV - Nachbarschaftsdichten beruhen wegen der überaus langen Rechenzeiten auf einer kleinen Datenbasis. Werden sie bei größerer Datengrundlage bestätigt?

5.Problem:

Woran liegt es, dass der Automatentyp esZA_1N4Z offensichtlich den größten Anteil an Klasse IV - Automaten besitzt?

6.Problem:

Gelten die Ergebnisse der vorliegenden Arbeit auch für esZA mit deutlich größeren Werten für n und z (d.h. z.B. für $n > 10$ und $z > 20$)?

7.Problem:

Bei den durchgeführten Untersuchungen hat es sich immer wieder herausgestellt, dass es eZA gibt, bei denen eine eindeutige subjektive (und damit auch algorithmische) Klassifikation schwierig, wenn nicht gar unmöglich ist. Ist es daher vielleicht erforderlich, das Wolframsche Klassifikationsschema, das lediglich auf vier Klassen beruht, zu überdenken und gegebenenfalls neu zu fassen?

Zum Abschluss eine eher erkenntnistheoretische Anmerkung:

Das ursprüngliche, mittlerweile Jahre zurückliegende, Ziel des Verfassers war es, eine Methode zu finden, wie man die Klasse, der ein eZA angehört, direkt dem Regelsatz des Automaten entnehmen kann, und zwar ohne die Raum-Zeit-Entfaltungen einzubeziehen! Dieses Ziel konnte nicht erreicht werden! Im Nachhinein, nach ungezählten Stunden intensiver Beschäftigung mit zellulären Automaten, erscheint das Scheitern verständlich: Komplexe Wechselwirkungssysteme gewinnen ihre Dynamik erst durch ihre Entfaltung. Natürlich hängt die Art der Dynamik einzig von den zugrundeliegenden Wechselwirkungsgesetzen und der Art der Vernetzung der das System konstituierenden Elemente ab. Trotzdem wäre niemand in der Lage, die Komplexität des Wechselwirkungssystems „Menschliches Gehirn“ auch nur annähernd vorherzusagen, wenn er lediglich die das System konstituierenden Elemente (nämlich die relativ einfach strukturierten Neuronen) kennen würde. Deshalb wurde es nötig, den Blick auf die Raum-Zeit-Entfaltungen von eZA zu lenken, um zu einem Klassifikationsalgorithmus zu gelangen.

Nun treffen Menschen die Entscheidung über die Klasse, der ein gegebener eZA angehört, anhand des Studiums seiner Raum-Zeit-Entfaltung. Diese Entscheidung wird oft in Bruchteilen einer Sekunde getroffen. Und die subjektiven Klassifikationen verschiedener Individuen decken sich in erstaunlicher Weise. Und es gab bisher keine andere Methode der Klassifizierung als diese: Willst du einen Automaten klassifizieren, so 'lasse ihn laufen' und sieh dir an, welche Muster sich ausbilden.

Hat man einen Klassifikationsalgorithmus zur Verfügung, so ist es nicht mehr notwendig, dass ein Mensch die Klassifikation persönlich vornimmt. Allerdings geht der vom Verfasser entwickelte Klassifikationsalgorithmus ähnlich vor wie ein menschlicher Klassifizierer: es werden (viele) verschiedene

Raum-Zeit-Entfaltungen protokolliert und anschließend analysiert. Und die Art der Analyse (Extraktion der rkZ und anschließende Suche nach bestimmten Mustern in der rkZ) ähnelt vielleicht den Mustererkennungsprozessen im menschlichen Gehirn. Es ist also denkbar, dass der Klassifikationsalgorithmus in gewisser Weise die Mustererkennungsstrukturen des menschlichen Gehirns modelliert.

6.Anhang

Im Folgenden werden wesentliche programmtechnische Einzelheiten im Zusammenhang mit dem Klassifikationsalgorithmus aufgelistet und dokumentiert. Die Programmierung erfolgte in der Programmiersprache C#. Als Plattform wurde das Betriebssystem Windows XP mit installiertem .NET Framework von Microsoft verwendet.

Folgende globale Variablenbedeutungen werden verwendet:

Der Regelsatz des esZA wird im Array mit Namen „f“ abgelegt. Das Array f ist allerdings eindimensional! Jede als $2n+1$ -Tupel gegebene mögliche Nachbarschaftszustandskonfiguration einer Zelle kann ja eindeutig als natürliche Zahl zur Basis z dargestellt werden. Vor Anwendung des Regelsatzes auf den Automaten muss also jede Nachbarschaftszustandskonfiguration in die zugehörige Zahl zur Basis z umgerechnet werden. Dieser Umweg wurde aus programmiertechnischen Gründen vorgenommen.

Die Anzahl der Zellen des esZA wird in der Variablen „WorldWidth“ abgelegt.

Die einzelnen Zellzustände sind im Array mit Namen „c“ abgelegt.

Die Variable „za“ gibt die Anzahl z der Zellzustände des Automaten an.

Die Variable „nas“ gibt die Anzahl n der Nachbarzellen auf einer Seite einer Zelle an.

Die Startbelegung des esZA mit zufällig gewählten Werten erfolgt mit der Routine void InitWorldArray():

```
void InitWorldArray() {
    Random Zufall = new Random((int)DateTime.Now.Ticks);
    for(int i=0;i<WorldWidth;i++) c[i] = Zufall.Next(0,za);
}
```

Die Berechnung der nachfolgenden Zellgeneration unter Zugrundelegung des Regelsatzes erledigt die Prozedur void new_Generation():

```
void new_Generation() {
    int z,h;
    for(z=0;z<WorldWidth;z++) {
        //Zunaechst erfolgt unter Beruecksichtigung des linken und rechten Randes
        //des Zellraumes die Umlagerung der Zellzustaende einer Zelle c[z] und ihrer
        //Nachbarzellen in die Koordinaten 0,...,2*nas eines Arrays namens „nb“:
        for(int k=0;k<2*nas+1;k++) {
            h = z-nas+k;
            if(h<0) {nb[k]=c[WorldWidth+h];}
            else {
                if(h<WorldWidth) {nb[k]=c[h];}
                else {nb[k]=c[h-WorldWidth];}
            }
        }
        //Danach erfolgt die Codierung der in nb abgelegten Nachbarschaftszustands-
        //konfiguration der Zelle c[z] als natuerliche Zahl zur Basis za in der
        //Koordinate z des Arrays namens „NCode“. Im Array p sind zum Zwecke der
        //Codierung die Potenzen za^0,za^1,...,za^(2*nas) abgelegt:
        NCode[z]=0;
        for(int k=0;k<2*nas+1;k++) NCode[z]=NCode[z]+nb[k]*p[k];
    }
    //Nun ist die Nachbarschaftszustandskonfiguration einer jeden Zelle als
    //natuerliche Zahl zur Basis za im Array NCode gespeichert.
    //Darauf kann nun der Regelsatz f angewendet werden, der die
    //nachfolgende Zellgeneration ermittelt und diese wieder im Array c ablegt:
    for(z=0;z<WorldWidth;z++) c[z]=f[NCode[z]];
}
```

Die Prozedur `int rkC_t(int Nmax)` ermittelt zu vorgegebenem N_{\max} den Wert von `rkC(t)` zum Zeitpunkt `t`. Der Wert von `rkC(t)` wird dann im Weiteren in der Koordinate `t` des Arrays namens „`rkC_t_Array`“ abgelegt, um die späteren Analysen zu ermöglichen.

```
int rkC_t(int Nmax) {
    double Zahlmax=(double) Math.Pow(za,WorldWidth);
    double Zahl=0.0;
    for(int j=0;j<WorldWidth;j++) {
        Zahl = Zahl + c[j] * (double) Math.Pow(za,j);
    }
    return Convert.ToInt16(Nmax*(Zahl/Zahlmax));
}
```

Eine Hilfsfunktion ist `int NumberOfrkC(int[] Feld,int Start,int End)`. Sie ermittelt in einem eindimensionalen Integer-Array „`Feld`“ die Anzahl der gespeicherten Werte, und zwar im Bereich von „`Start`“ bis „`End`“.

```
int NumberOfrkC(int[] Feld,int Start,int End) {
    int erg=1;
    for(int j=Start+1;j<=End;j++) {
        if(isElementOf(Feld[j],Feld,Start,j-1)==false) erg++;
    }
    return erg;
}
```

Diese Hilfsfunktion ruft ihrerseits die Hilfsfunktion `bool isElementOf(int obj,int[] Feld,int Start,int End)` auf, die testet, ob eine vorgegebene Zahl „`obj`“ in einem eindimensionalen Integer-Array „`Feld`“ im Bereich von „`Start`“ bis „`End`“ abgelegt wurde. Falls ja, gibt diese Funktion „`true`“ zurück, ansonsten „`false`“:

```
bool isElementOf(int obj,int[] Feld,int Start,int End) {
    bool erg=false;
    int pt=Start;
    while(pt<=End) {
        erg=(obj==Feld[pt]);
        if(erg==true) pt=End;
        pt++;
    }
    return erg;
}
```

Klasse I - Automaten haben die Eigenschaft, nach nur wenigen Zellgenerationen in einen einheitlichen Zellzustand zu gelangen (d.h. alle Zellen befinden sich dann im gleichen Zustand). Die nachfolgende Funktion `bool all_cells_are_equal(int after_n_Generations,int tests)` überprüft genau dies. Die Überprüfung wird meistens nicht nur ein einziges Mal, sondern „`tests`“ Mal durchgeführt, denn auch Klasse II - Automaten haben manchmal die Eigenschaft, dass sie bei zufällig gewählten Startbelegungen nach einigen Zellgenerationen in einen einheitlichen Zellzustand gelangen. Erst wenn alle diese Überprüfungen positiv ausgehen, wird am Ende „`true`“ ausgegeben, ansonsten „`false`“. Je größer die Zahl „`after_n_Generations`“ ist, desto größer ist die Sicherheit:

```
bool all_cells_are_equal(int after_n_Generations,int tests) {
    bool erg=true;
    for(int i=0;i<tests;i++) {
        //Zufaellige Startbelegung des esZA:
        InitWorldArray();
        //after_n_Generations - malige Entfaltung des esZA:
        for(int j=0;j<after_n_Generations;j++) new_Generation();
        //Ueberpruefe nun, ob sich alle Zellen im gleichen Zustand befinden:
    }
```

```

    for(int j=0;j<WorldWidth;j++) {
        if(c[j]!=c[0]) {
//falls ein Zellzustand nicht mit c[0] uebereinstimmt, wird die
//gesamte Ueberpruefung abgebrochen und false ausgegeben.
            erg=false;
            break;
        }
    }
    if(erg==false) break;
}
return erg;
}

```

Nun der zentrale Klassifikationsalgorithmus `string Classification(int tWindow,int Tests)`, der den Kernalgorithmus umfasst. Der Klassifikationsalgorithmus als Hauptalgorithmus ruft den Kernalgorithmus „Tests“ Mal auf. Die Länge des Zeitfensters (siehe 3.Abschnitt) wird mit dem Parameter „tWindow“ übergeben. Der Klassifikationsalgorithmus gibt die algorithmisch ermittelte Klasse in Form einer Zeichenkette („I“, „II“, „III“ oder „IV“) zurück:

```

string Classification(int tWindow,int Tests) {
    string erg;
//Im Array ClassArray werden die Ergebnisse des Kernalgorithmus eingetragen.
    int[] ClassArray = new int[5];
//Dieses Array wird zunaechst mit 0 initialisiert:
    for(int j=1;j<5;j++) ClassArray[j]=0;
//Variable fuer den Start einer evtl. erkannten zeitlichen
//Periode von rkC(t):
    int PeriodStart=1000;
//Hilfsvariablen fuer die minimale Anzahl von rkC(t)-Werten
//in einem Zeitfenster vorgegebener Länge:
    int NORkC_help;
    int NORkC_min;
//Die Protokollierung der rkC(t)-Werte ueber einen Zeitraum von
//time=1000 Zeitschritten hat sich als ausreichend lang erwiesen:
    int time = 1000;
//Eine Weltbreite von 200 ist fuer die Analysen groß genug:
    WorldWidth=200;
//Jetzt startet der Kernalgorithmus.Er wird Tests Mal aufgerufen:
    for(int ts=0;ts<Tests;ts++) {
//Zuerst die zufällige Starbelegung des esZA:
        InitWorldArray();
//Berechnung von rkC(0) und Speicherung in der 0.Koordinate
//des Arrays rkC_t_Array:
        rkC_t_Array[0]=rkC_t(5000);
//(time-1)-malige Entfaltung des esZA, Berechnung der rkC(t)-Werte
//und Ablegen der Ergebnisse im Array rkC_t_Array:
        for(int j=1;j<time;j++) {
            new_Generation(WorldWidth);
            rkC_t_Array[j] = rkC_t(5000);
        }
//Es wird ueberprueft, ob die rkZ eine zeitliche Periode hat, die hinreichend
//klein (kleiner oder gleich des eineinhalbfachen der Weltbreite) ist.
//Falls ja, wird der Startzeitpunkt dieser Periode ermittelt und in der
//Variablen
//PeriodStart abgelegt.
//Falls keine solche Periode vorliegt, wird PeriodStart auf time gesetzt.
        PeriodStart=time;
        if((rkC_t_Array[798]==rkC_t_Array[998])||(rkC_t_Array[799]==rkC_t_Array[999]))
        {
            PeriodStart = 798;
        }
    }
while((rkC_t_Array[PeriodStart]==rkC_t_Array[PeriodStart+200])&&(PeriodStart>0))
    PeriodStart--;
}
if(PeriodStart==time) {

```

```

if((rkC_t_Array[698]==rkC_t_Array[998])||(rkC_t_Array[699]==rkC_t_Array[999])) {
    PeriodStart = 698;
}

while((rkC_t_Array[PeriodStart]==rkC_t_Array[PeriodStart+300])&&(PeriodStart>0))
    PeriodStart--;
}
}
if(PeriodStart<time) {
//Falls eine solche Periode vorliegt, wird ueberprueft, ob sie nach weniger
//als 30 Generationen beginnt:
    if(PeriodStart<30) {
//Wenn ja, wird auf Klasse I getestet:
        if(all_cells_are_equal(50,5)) {ClassArray[1]++;}
//Falls nicht Klasse I vorliegt, wird Klasse II konstatiert:
        else {ClassArray[2]++;}
    }
//Wenn die Periode spaeter als nach 30 Generationen startet, wird
//die Klasse IV konstatiert:
    else {ClassArray[4]++;}
}
else {
//Falls keine Periode vorliegt, wird ein Zeitfenster der Laenge tWindow
//ueber das Array rkC_t-Array gezogen und die minimale Anzahl
//der rkC(t)-Werte ermittelt, die in einem solchen Zeitfenster gezaehlt werden.
//Diese minimale Anzahl wird in der Variablen NORkC_min gespeichert.
    NORkC_min=100;
    for(int j=0;j<1000-Window;j++) {
        NORkC_help=NumberOfrkC(rkC_t_Array,j,j+tWindow);
        if(NORkC_help<NORkC_min) NORkC_min=NORkC_help;
    }
//Betraegt die minimale Anzahl hoechstens 9, so wird der Automat
//der Klasse IV zugerechnet:
    if(NORkC_min<=9) {ClassArray[4]++;}
//Ansonsten wird die Klasse III konstatiert:
    else {ClassArray[3]++;}
}
}
//Hier ist der Kernalgorithmus beendet.
//Es wird nun im Array ClassArray nachgesehen, welche Klasse am haeufigsten
//erkannt wurde. Dann wird die entsprechende Klasse in der Variablen
//namens AutomataClass gespeichert und als Ergebnis ausgegeben:
int ClassIndex=1;
for(int j=2;j<5;j++) {
    if(ClassArray[j]>ClassArray[ClassIndex]) ClassIndex=j;
}
if(ClassIndex==1) AutomataClass="I";
if(ClassIndex==2) AutomataClass="II";
if(ClassIndex==3) AutomataClass="III";
if(ClassIndex==4) AutomataClass="IV";
//Eine Besonderheit stellt die Klasse II dar:
//falls die vom Kernalgorithmus am haeufigsten erkannte Klasse die
//Klasse I ist, aber dennoch mindestens ein Mal die Klasse II erkannt
//wurde, kann der Automat nicht mehr der Klasse I angehoren, sondern
//muss der Klasse II zugerechnet werden:
if((AutomataClass=="I") && (ClassArray[2]>0)) AutomataClass="II";
erg=AutomataClass;
//Ausgabe des Ergebnisses als Zeichenkette („I“, „II“, „III“ oder „IV“):
return erg;
}

```

Die hier dokumentierten Programmroutinen stellen natürlich nur einen kleinen Teil der Prozeduren und Funktionen dar, die für ein lauffähiges C#-Programm erforderlich sind. Auf die Dokumentation dieser Programmteile wird verzichtet, da es in dieser Arbeit um den Klassifikationsalgorithmus geht und nicht um programmtechnische Details. Insbesondere wird hier nicht beschrieben, mit welchen Algorithmen die Dezimalcodierung des Regelsatzes bei sehr großen natürlichen Zahlen (mit bis zu mehreren hundert Stellen!) erfolgt.

Bei Bedarf können diese anderen Routinen vom Verfasser bezogen werden (EMail: martin.sollert@ewetel.net).

7.Literaturverzeichnis

[D]A.Deutsch (Hrsg.), "Muster des Lebendigen", vieweg Verlag, Braunschweig, Wiesbaden 1994

[GeSch]M. Gerhardt, H.Schuster, "Das digitale Universum", vieweg Verlag, Braunschweig, Wiesbaden 1995

[L]S.Levy, "KL-Künstliches Leben aus dem Computer", Verlag Droemer Knaur, München 1993

[Wo1]S.Wolfram, "Cellular Automata and Complexity", Addison-Wesley, Reading 1994

[Wo2]S. Wolfram, "A new kind of science", Wolfram Media, Winnipeg 2002

[Wu]A.Wuensche, "Classifying Cellular Automata Automatically", Santa Fe Institute working paper 98-02-018, 1998

[WuLe]A.Wuensche, Mike Lesser, "The Global Dynamics of Cellular Automata", Addison-Wesley, Reading 1992, ISBN 0-201-55740-1