

Zusammenfassung

Bevor wir in den folgenden Kapiteln unterschiedlichste Simulationsmöglichkeiten mit zellulären Automaten vorstellen, fassen wir hier einige Grundlagen und erklärende Beispiele zusammen. Zunächst beginnen wir in Abschn. 1.1 mit einer knappen historischen Entwicklung von zellulären Automaten, bevor wir in Abschn. 1.2 eine allgemeine Beschreibung präsentieren, an die wir uns in allen Modellen in diesem Buch halten werden. Weiterführende Bemerkungen zur Beschreibung werden anschließend in Abschn. 1.3 diskutiert. In den folgenden Abschn. 1.4 und 1.5 veranschaulichen wir die zuvor eingeführten Definitionen und Bezeichnungen an einem eindimensionalen sowie an einem zweidimensionalen Modell. Diese grundlegenden Beispiele sind zugleich auch historisch bedeutende zelluläre Automaten, die überhaupt erst das Interesse an dieser Simulationsmöglichkeit geweckt haben.

1.1 Einleitung

Wie im Vorwort bereits ausführlich beschrieben, dienen zelluläre Automaten zur agenten-basierten Modellierung und Simulation verschiedenster Sachverhalte. Die Entstehung des Lebens, Modelle im Verkehrswesen, populationsdynamische Entwicklungen oder die Ausbreitung von Epidemien stellen dabei nur eine kleine Auswahl der unzähligen Szenarien dar, die sich mit zellulären Automaten simulieren lassen.

Mit der Verbreitung der Computer veröffentlichte John von Neumann eine der grundlegenden Arbeiten zu zellulären Automaten, s. Neumann (1966). Maßgeblich vorangetrieben wurde die Entwicklung und Erforschung der zellulären Automaten durch Stephen Wolfram in einer ganzen Reihe von Arbeiten, s. z. B. Wolfram (1983), Wolfram (1984a), Wolfram (1984b), Wolfram (1986a), Wolfram (1986b), Wolfram (1994) und Wolfram (2002). Im Laufe der Zeit haben zelluläre Automaten in einer Vielzahl von Fachgebieten Interesse gefunden, sodass mitunter sehr komplexe und hochinteressante Arbeiten entstanden sind.

In diesem Buch stellen wir vergleichsweise einfache Modelle aus unterschiedlichen Bereichen wie z. B. der Physik, der Biologie, dem Verkehrswesen oder den Ingenieurwissenschaften vor. Einige dieser Modelle sind teilweise auch in den Lehrbüchern Margolus und Toffoli (1987), Gerhardt und Schuster (1995), Chaudhuri et al. (1997), Wolfram (2002) und Schiff (2008) zu finden. Unser Fokus liegt einerseits darauf, alle Modelle so einfach wie nur möglich und zugleich anschaulich darzustellen. Andererseits wollen wir zeigen, dass sich durch einfache Simulationen bereits realitätsnahe Ergebnisse erzielen lassen, sodass etwa das Verhalten konkurrierender Spezies oder physikalische Gesetze verifiziert werden können.

Im Gegensatz zu **deterministischen** Modellen, bei denen mit gleichen Eingangsdaten exakt die gleichen Ergebnisse erzielt werden, untersuchen wir in den folgenden Kapiteln stets **randomisierte** Modelle, bei denen der Zufall eine ganz wesentliche Rolle spielt. Wir werden sehen, wie sich faszinierende Ergebnisse gerade aufgrund zufälliger Prozesse erreichen lassen.

Bevor wir jedoch mit der Präsentation unterschiedlicher Modelle beginnen, führen wir in diesem Kapitel zunächst eine allgemeine Beschreibung von zellulären Automaten ein. Die Beschreibung im folgenden Abschn. 1.2 kann je nach Vorwissen auf den ersten Blick etwas abstrakt oder schwer verständlich wirken, der generelle Aufbau und der Ablauf von zellulären Automaten sollten dann aber in den Beispielen der Abschn. 1.4 und 1.5 leicht nachvollziehbar sein.

1.2 Zelluläre Automaten

Wie bereits in der Einleitung beschrieben, lassen sich **zelluläre Automaten** zur Modellierung und Simulation dynamischer Systeme verwenden. Dabei werden diskrete Zeitschritte

$$t_0 \rightarrow t_1 \rightarrow t_2 \rightarrow t_3 \rightarrow t_4 \rightarrow t_5 \rightarrow t_6 \rightarrow \dots$$

betrachtet, wobei t_0 den Startzeitpunkt der Simulation festlegt. Jede Simulation mit einem zellulären Automaten kann nun eindeutig durch die folgenden sechs Größen bzw. Eigenschaften definiert werden.

1.2.1 Spielfeld

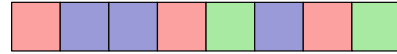
Zunächst wird ein diskreter Raum benötigt, in dem die Simulation durchgeführt wird. In diesem Buch beschränken wir uns auf **eindimensionale** und **zweidimensionale** Räume, welche wir als **Spielfelder**¹  bezeichnen werden.

Ein eindimensionales Spielfeld besteht dabei aus einer Reihe von n **Pixeln**, nämlich

$$x_1, x_2, x_3, x_4, \dots, x_n$$

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
-------	-------	-------	-------	-------	-------	-------	-------

(a) Spielfeld und Pixel



(b) Spielfeld mit Zuständen

Abb. 1.1 Beispiel eines eindimensionalen Spielfeldes mit $n = 8$ Pixeln. In (a) ist das Spielfeld mit der Bezeichnung der Pixel zu sehen. In (b) wurde jedem Pixel ein Zustand zugeordnet, wobei die Zustandsmenge Q aus den $s = 3$ Zuständen q_1 , q_2 und q_3 besteht, denen *Rot*, *Grün* und *Blau* zugeordnet ist. Ein rotes Pixel bedeutet beispielsweise, dass sich das Pixel im Zustand q_1 befindet

oder kürzer x_i für $i = 1, \dots, n$, s. Abb. 1.1a. Jedes Pixel nimmt zu jedem Zeitpunkt einen Zustand an, wie wir später genauer erklären werden.

Ein zweidimensionales Spielfeld besteht aus einem rechteckigen Raster oder Gitter mit $m \times n$ **Pixeln**, nämlich

$$x_{ij} \quad \text{für} \quad i = 1, \dots, m \quad \text{und} \quad j = 1, \dots, n,$$

wobei jedes Pixel wieder einen Zustand einer Zustandsmenge einnehmen kann, s. Abb. 1.2a.

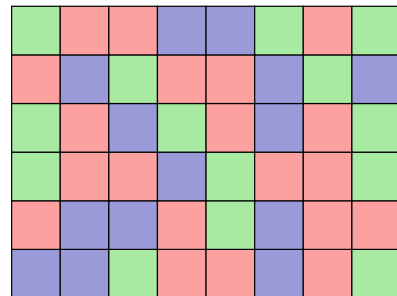
1.2.2 Zustände

Jedes Pixel im Spielfeld nimmt zu jedem Zeitpunkt genau einen **Zustand**  ein. Dazu definieren wir die **Zustandsmenge**

$$Q = \{q_1, \dots, q_s\},$$

x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}
x_{21}	x_{22}	x_{23}	x_{24}	x_{25}	x_{26}	x_{27}	x_{28}
x_{31}	x_{32}	x_{33}	x_{34}	x_{35}	x_{36}	x_{37}	x_{38}
x_{41}	x_{42}	x_{43}	x_{44}	x_{45}	x_{46}	x_{47}	x_{48}
x_{51}	x_{52}	x_{53}	x_{54}	x_{55}	x_{56}	x_{57}	x_{58}
x_{61}	x_{62}	x_{63}	x_{64}	x_{65}	x_{66}	x_{67}	x_{68}

(a) Spielfeld und Pixel



(b) Spielfeld mit Zuständen

Abb. 1.2 Beispiel eines zweidimensionalen Spielfeldes mit $m \times n = 6 \times 8 = 48$ Pixeln. In (a) ist das Spielfeld mit der Bezeichnung der Pixel zu sehen. In (b) wurde jedem Pixel ein Zustand zugeordnet, wobei die Zustandsmenge Q aus den $s = 3$ Zuständen q_1 , q_2 und q_3 besteht, denen *Rot*, *Grün* und *Blau* zugeordnet ist. Ein *rotes* Pixel bedeutet beispielsweise, dass sich das Pixel im Zustand q_1 befindet

welche aus den s Zuständen q_1 bis q_s besteht. Zur späteren Veranschaulichung der Simulation ordnen wir jedem Zustand genau eine Farbe zu, s. Abb. 1.1b und 1.2b.

1.2.3 Nachbarschaft

Jedes Pixel besitzt eine **Nachbarschaft** 3 , wobei für jedes Pixel die Nachbarschaft nach den gleichen Regeln bestimmt wird. Dabei besteht eine Nachbarschaft aus dem jeweiligen Pixel selbst sowie einigen Pixeln in der Nähe von ihm.

Beispiele von Nachbarschaften werden wir im folgenden Abschn. 1.3.1 genauer vorstellen.

1.2.4 Schritte

Bislang haben wir nur grundlegende Eigenschaften festgelegt und die Rolle der Zeitpunkte t_k vernachlässigt. Die wichtigste Eigenschaft von zellulären Automaten ist jedoch die folgende:

Der Zustand eines Pixels zum Zeitpunkt t_{k+1} ergibt sich aus den Zuständen der Pixel in seiner Nachbarschaft zum Zeitpunkt t_k .

Dabei wird der Zustand eines Pixels im jeweils folgenden Zeitschritt durch **Regeln** 4  bestimmt. Wir unterscheiden zwischen zwei Arten von Regeln:

1. Regeln, bei denen der Zustand eines Pixels durch die Zustände der Pixel in seiner Nachbarschaft festgelegt werden. Bei derartigen Regeln kann sich nur der Zustand des jeweiligen Pixels ändern.
2. Regeln, bei denen Pixel innerhalb einer Nachbarschaft miteinander interagieren. Dabei ist es möglich, dass sich die Zustände mehrerer Pixel innerhalb der Nachbarschaft ändern.

Beim Übergang aller Pixel im Spielfeld von einem Zeitpunkt t_k in den folgenden Zeitpunkt t_{k+1} unter Verwendung einer Regel vom Typ (1) oder nach einer zuvor festgelegten Anzahl von Anwendungen einer Regel vom Typ (2) sprechen wir von einem **Schritt**. Weiter bezeichnen wir die **Konfiguration** des Spielfeldes zum Zeitpunkt t_k als **Generation** G_k .

In den folgenden Kapiteln untersuchen wir stets zelluläre Automaten, bei denen der Zufall bei der Berechnung eines Schrittes eine wesentliche Rolle spielt.

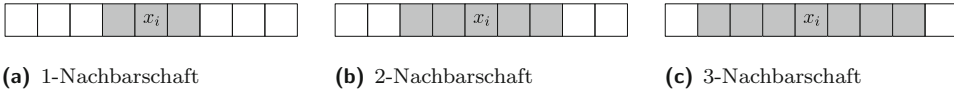


Abb. 1.3 Beispiele von Nachbarschaften in eindimensionalen Spielfeldern. Die Nachbarschaft von Pixel x_i ist jeweils in *Grau* dargestellt

1.2.5 Randbedingungen

Zur Festlegung der Regeln ist stets auf **Randbedingungen** 5 zu achten. Wie bereits beschrieben, ergibt sich der neue Zustand eines Pixels aus seiner Nachbarschaft. Wenn das Pixel nun aber am Rand des Spielfeldes liegt, kann es vorkommen, dass seine Nachbarschaft nicht komplett im Spielfeld enthalten ist. Für diese Fälle müssen besondere Regeln gefunden werden, nämlich genau die Randbedingungen, s. Abschn. 1.3.2.

1.2.6 Startkonfiguration

Damit ist ein zellulärer Automat im Grunde vollständig beschrieben. Was zur Simulation noch fehlt, sind jeweils Überlegungen zur **Startkonfiguration** 6 im Zeitpunkt t_0 , bevor die Simulation durchgeführt werden kann. Darunter verstehen wir die Festlegung der ersten Generation G_0 , also die initiale Zuweisung eines Zustandes für alle Pixel.

1.3 Bemerkungen

Bevor wir die Beschreibung von zellulären Automaten an zwei Beispielen genauer studieren, wollen wir an dieser Stelle noch auf zwei Punkte näher eingehen.

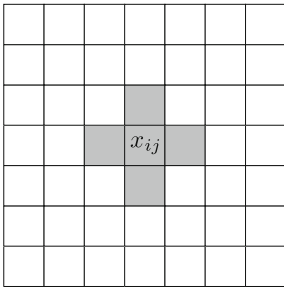
1.3.1 Nachbarschaften

1.3.1.1 Nachbarschaften in eindimensionalen Spielfeldern

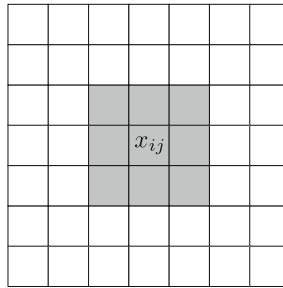
In eindimensionalen Spielfeldern ist die Definition von Nachbarschaften relativ einfach. Mit einer natürlichen Zahl z besteht die z -Nachbarschaft von x_i aus dem Pixel x_i selbst sowie aus den z jeweils links und rechts benachbarten Pixel, s. Abb. 1.3.

1.3.1.2 Nachbarschaften in zweidimensionalen Spielfeldern

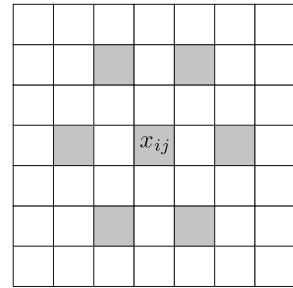
In zweidimensionalen Spielfeldern gibt es sehr viel mehr Möglichkeiten zur Definition von Nachbarschaften. In Abb. 1.4a ist die **Von-Neumann-Nachbarschaft** zu sehen, die aus x_{ij} selbst sowie vier weiteren Pixeln besteht. Bei der **Moore-Nachbarschaft** in Abb. 1.4b erhalten wir insgesamt neun Pixel, also x_{ij} und acht weitere Pixel.



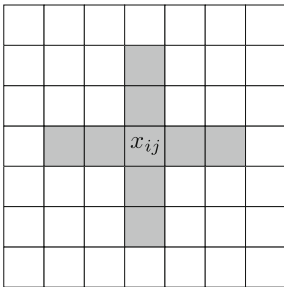
(a) Von-Neumann-Nachbarschaft



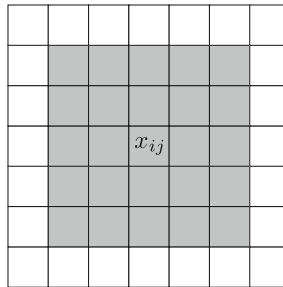
(b) Moore-Nachbarschaft



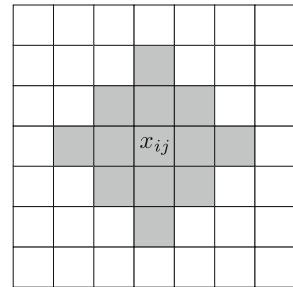
(c) Hexagonale Nachbarschaft



(d) 2-Von-Neumann-Nachbarschaft



(e) 2-Moore-Nachbarschaft



(f) 2-Nachbarschaft

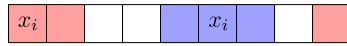
Abb. 1.4 Beispiele von Nachbarschaften in zweidimensionalen Spielfeldern. Die Nachbarschaft von Pixel x_{ij} ist jeweils in *Grau* dargestellt

Die Definition einer **hexagonalen Nachbarschaft** mit x_{ij} sowie sechs weiteren Pixeln ist ein wenig komplizierter. Wir haben uns für die in Abb. 1.4c gezeigte Variante entschieden. Diese hat den Vorteil, dass sich die hexagonale Nachbarschaft auch auf einem rechteckigen Spielfeld sehr gut darstellen lässt. Ein großer Nachteil ist jedoch, dass dabei nur jedes vierte Pixel tatsächlich von Interesse ist. Mit anderen Worten, nur ein Viertel aller Pixel auf dem Spielfeld interagiert miteinander, sodass hier viel Speicherplatz verschwendet wird.

Weitere Möglichkeiten zweidimensionaler Nachbarschaften sind den Abb. 1.4d und 1.4e zu entnehmen.

1.3.2 Randbedingungen

Wenn ein Pixel am Rand des Spielfeldes liegt, sodass seine Nachbarschaft nicht komplett im Spielfeld enthalten ist, müssen unter Umständen Randbedingungen definiert werden. Wir stellen hier zwei Möglichkeiten vor.



(a) 1-Nachbarschaft



(b) 2-Nachbarschaft

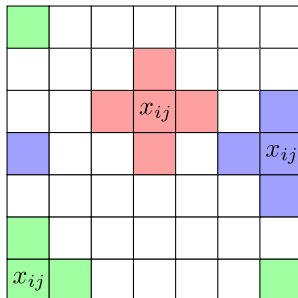
Abb. 1.5 Veranschaulichung der periodischen Randbedingungen bei eindimensionalen Spielfeldern. Die *farbigen* Pixel sind jeweils die Nachbarschaft von x_i . Die jeweils *rote* Nachbarschaft ist nicht komplett im Spielfeld enthalten. Bei periodischen Randbedingungen wird die Nachbarschaft nun durch Pixel vom anderen Rand des Spielfeldes ergänzt

1.3.2.1 Abgeschlossene Randbedingungen

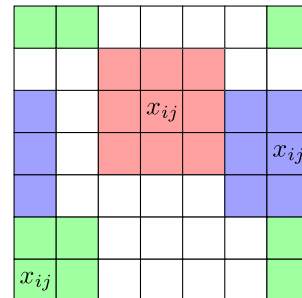
Wir sprechen von *abgeschlossenen Randbedingungen*, wenn wir folgende Regeln verwenden: Angenommen, die Nachbarschaft eines Pixels ist nicht komplett im Spielfeld enthalten, dann wird sich der Zustand des Pixels niemals ändern. Mit anderen Worten, von Schritt zu Schritt wird der Zustand des Pixels ohne Beachtung der Regeln stets übernommen.

1.3.2.2 Periodische Randbedingungen

Sehr viel interessanter ist die Definition von *periodischen Randbedingungen*. Wenn in diesem Fall die Nachbarschaft eines Pixels nicht komplett im Spielfeld enthalten ist, ergänzen wir die fehlenden Pixel der Nachbarschaft durch Pixel am anderen Rand des Spielfeldes. Da eine genaue Definition in Worten sehr schwierig ist, haben wir in Abb. 1.5 die periodischen Randbedingungen für eindimensionale Spielfelder sowie in Abb. 1.6 für zweidimensionale Spielfelder genau dargestellt.



(a) Von-Neumann-Nachbarschaft



(b) Moore-Nachbarschaft

Abb. 1.6 Veranschaulichung der periodischen Randbedingungen bei zweidimensionalen Spielfeldern. Die *farbigen* Pixel sind jeweils die Nachbarschaft von x_{ij} . In (a) ist die Von-Neumann-Nachbarschaft dargestellt. Auch hier werden die Nachbarschaften, die nicht komplett im Spielfeld enthalten sind, durch entsprechende Pixel ergänzt. Besonders interessant ist der Fall, bei dem x_{ij} in einer Ecke des Spielfeldes liegt. Hier wird die Nachbarschaft durch Pixel in anderen Ecken des Spielfeldes ergänzt. Die gleichen Fälle für die Moore-Nachbarschaft sind in (b) dargestellt

Tab. 1.1 Übersicht der beiden Zustände im elementaren zellulären Automaten

Zustand	Farbe	Beschreibung
q_1		Pixel nicht besetzt
q_2		Pixel besetzt

1.4 Eindimensionales Universum

Nach der bislang theoretischen Einführung von zellulären Automaten sehen wir uns nun erste Beispiele an.

1.4.1 Elementarer zellulärer Automat

Wir untersuchen den *elementaren zellulären Automaten*, welcher ursprünglich von Stephen Wolfram in Wolfram (1983) veröffentlicht wurde und häufig als der einfachste nicht-triviale zelluläre Automat bezeichnet wird.

Es handelt sich beim elementaren zellulären Automaten um ein eindimensionales Spielfeld  bestehend aus n Pixeln. Jedes Pixel kann zwei Zustände  annehmen, Pixel nicht besetzt (q_1) und Pixel besetzt (q_2), s. Tab. 1.1. Pixel, die nicht besetzt sind, werden wir in Hellgrau darstellen und Pixel, die besetzt sind, in Schwarz.

Weiterhin wählen wir eine 1-Nachbarschaft , sodass die Nachbarschaft jeden Pixels aus drei Pixeln besteht, s. Abb. 1.3a. Als Nächstes müssen wir eine Beschreibung der Schritte angeben und verwenden dazu Regeln vom Typ (1), s. Abschn. 1.2.4.

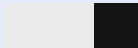
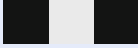
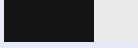
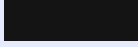
Da in diesem Beispiel jedes Pixel nur zwei Zustände annehmen kann und jede Nachbarschaft aus drei Pixeln besteht, gibt es nur $2^3 = 8$ mögliche Regeln, die einen Schritt eindeutig beschreiben. Tabelle 1.2 zeigt diese acht Regeln . Auf der linken Seite wird jeweils die Nachbarschaft eines Pixels x_i zum Zeitpunkt t_k gezeigt. Der Zustand des Pixels x_i zum folgenden Zeitpunkt t_{k+1} ist schließlich auf der rechten Seite dargestellt. Unabhängig von Tab. 1.2 hätte man die Regeln in Worten auch folgendermaßen formulieren können:

Das Pixel x_i ist zum Zeitpunkt t_{k+1} genau dann besetzt, wenn zum Zeitpunkt t_k genau eines der beiden Pixel links und rechts von x_i besetzt ist.

Neben den eindeutigen Regeln ist ein Schritt aber erst dann vollständig beschrieben, nachdem Randbedingungen festgelegt wurden. Wir wählen an dieser Stelle periodische Randbedingungen , s. Abschn. 1.3.2, sodass damit jeder Schritt eindeutig festgelegt ist.

Als erstes Beispiel betrachten wir ein Spielfeld bestehend aus $n = 15$ Pixeln. Die Startkonfiguration  bzw. Startgeneration G_0 sei so gewählt, dass alle Pixel bis auf das mittlere Pixel x_8 nicht besetzt sind. In Abb. 1.7 sind nach Anwendung der zuvor beschriebenen Regeln die ersten sieben Schritte, also die Generationen G_1 bis G_7 , in der *Pixel-Zustands-Ansicht*

Tab. 1.2 Übersicht aller Regeln im elementaren zellulären Automaten

Nachbarschaft von x_i zum Zeitpunkt t_k		Zustand von x_i zum Zeitpunkt t_{k+1}
	→	
	→	
	→	
	→	
	→	
	→	
	→	
	→	

dargestellt. Dabei ist die Abbildung so zu verstehen, dass das Spielfeld zum Zeitpunkt t_k der Zeile $k + 1$ entspricht. Die erste Zeile entspricht somit der Startkonfiguration G_0 , die zweite Zeile entspricht der Generation G_1 und so weiter.

 In Abb. 1.8 ist das gleiche Beispiel nochmals dargestellt, nun allerdings mit einem 801 Pixel großen Spielfeld.

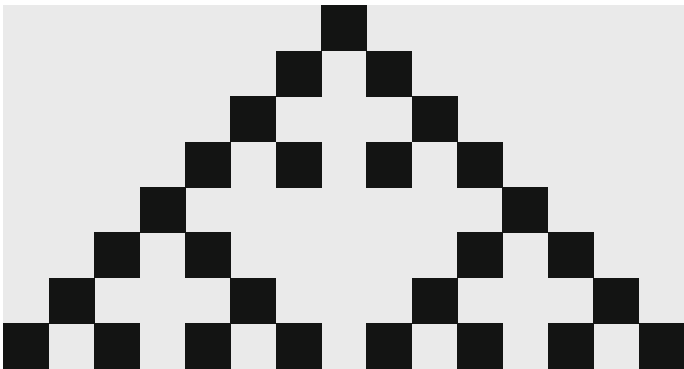


Abb. 1.7 Veranschaulichung des elementaren zellulären Automaten auf einem Spielfeld mit $n = 15$ Pixeln. In der ersten Zeile ist die Startkonfiguration G_0 zu sehen, in der nur ein Pixel besetzt ist. Durch Anwendung der in Tab. 1.2 beschriebenen Regeln ergeben sich die Generationen G_1 bis G_7 in den Zeilen 2 bis 8

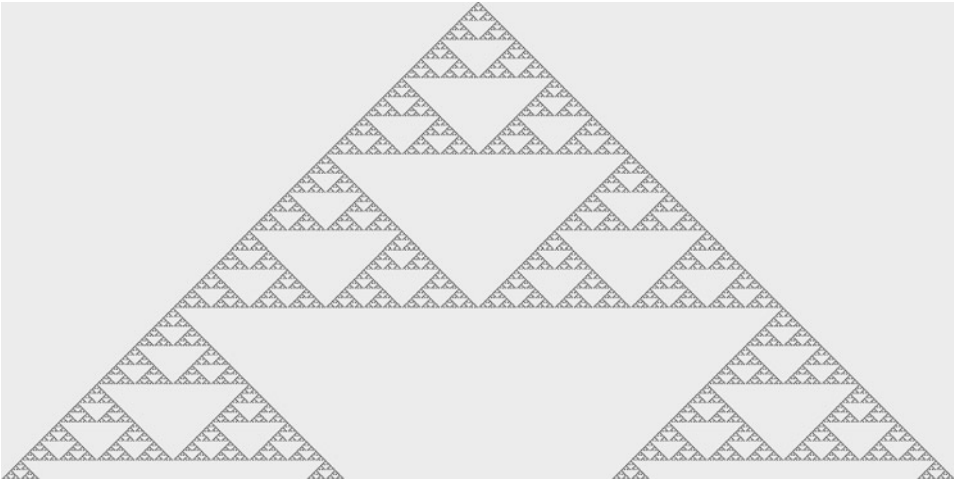


Abb. 1.8 Veranschaulichung des elementaren zellulären Automaten auf einem Spielfeld mit $n = 801$ Pixeln. Dargestellt sind die ersten 400 Generationen

An dieser Stelle sei darauf hingewiesen, dass es sich bislang um einen rein deterministischen zellulären Automaten handelt. Weder bei der Durchführung eines Schrittes noch bei der Startkonfiguration spielte der Zufall eine Rolle.

Weiterhin lassen sich die Regeln aus Tab. 1.2 natürlich beliebig variieren, wir werden es aber zunächst bei diesem einen Beispiel belassen.

1.4.2 Eindimensionales Universum

Nun wollen wir den elementaren zellulären Automaten ein wenig erweitern, um die Entstehung des Lebens auf der Erde nach dem Urknall so einfach wie möglich zu simulieren. Daher wird das folgende Beispiel auch als **eindimensionales Universum** bezeichnet.

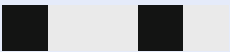
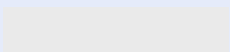
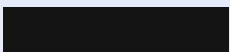
Wieder untersuchen wir ein eindimensionales Spielfeld 1×100 mit periodischen Randbedingungen 5×4 sowie der Zustandsmenge 2×2 aus Tab. 1.1, wobei ein besetztes Pixel die Anwesenheit eines Lebewesens veranschaulicht. Anders als zuvor wählen wir nun eine 2-Nachbarschaft 3×1 , s. Abb. 1.3b.

Die Schritte von einer bestehenden Generation zur folgenden Generation lassen sich durch eine einzige Regel 4×2 vom Typ (1) erklären:

Wenn entweder zwei oder vier Pixel aus der Nachbarschaft von x_i besetzt sind, dann ist x_i in der folgenden Generation besetzt, ansonsten nicht.

Tabelle 1.3 veranschaulicht diese Regel exemplarisch an einigen Beispielen.

Tab. 1.3 Beispiele der Regel im eindimensionalen Universum

Nachbarschaft von x_i zum Zeitpunkt t_k		Zustand von x_i zum Zeitpunkt t_{k+1}
	→	
	→	
	→	
	→	
	→	
	→	
	→	
	→	

 In Abb. 1.9 sind sechs Beispiele des eindimensionalen Universums auf einem 400 Pixel großen Spielfeld veranschaulicht, bei denen die Startkonfiguration **6**  jeweils nach folgendem Prinzip zufällig ermittelt wurde: Jedes Pixel der Generation G_0 ist mit einer Wahrscheinlichkeit von $\frac{1}{2}$ besetzt bzw. nicht besetzt.

Wie zuvor auch ist die Startkonfiguration G_0 in der jeweils obersten Zeile zu finden. Die folgenden Generationen sind chronologisch von oben nach unten angeordnet. Wie Abb. 1.9 zu entnehmen ist, entwickeln sich abhängig von der zufälligen Startkonfiguration äußerst interessante und vielseitige Strukturen.

1.4.3 Einfluss von Randbedingungen

Abschließend wollen wir am Beispiel des eindimensionalen Universums den Einfluss der Randbedingungen verdeutlichen. Zur wieder zufällig generierten, aber exakt identischen Startkonfiguration berechnen wir das eindimensionale Universum einmal mit abgeschlossenen und einmal mit periodischen Randbedingungen **5** , s. Abschn. 1.3.2.

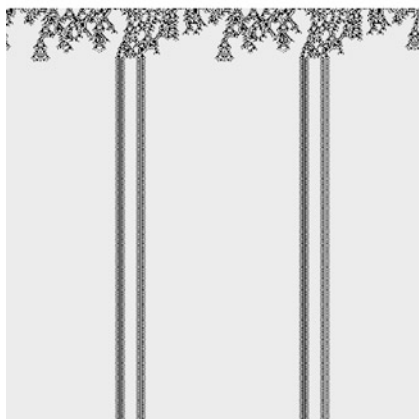
Ein Beispiel hierzu kann Abb. 1.10 entnommen werden. Es ist zu erkennen, dass die Wahl der Randbedingungen schon in diesem sehr einfachen Modell einen recht großen Einfluss auf die gesamte Simulation nehmen kann. Daher ist es bei der Modellierung mit zellulären Automaten äußerst wichtig, sich über Randbedingungen Gedanken zu machen.



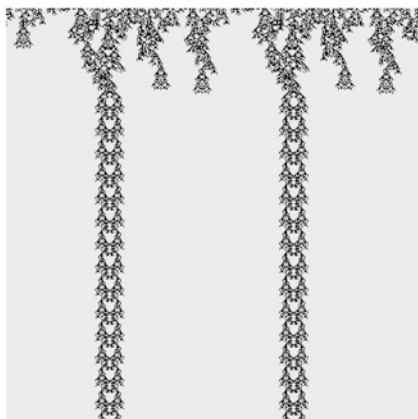
(a) Population stirbt schnell aus



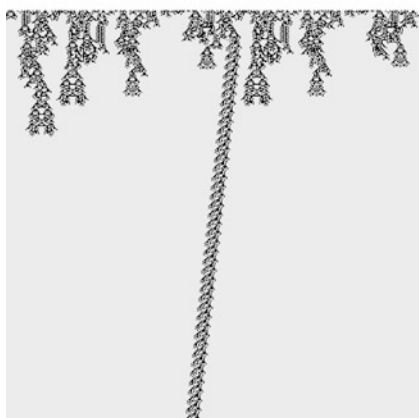
(b) Population stirbt langsam aus



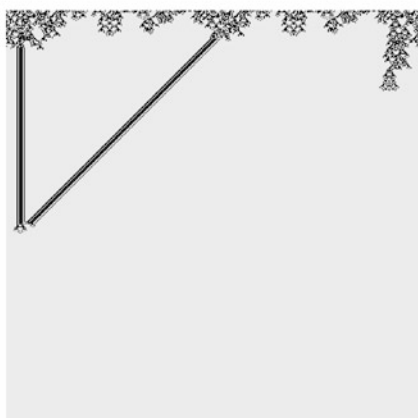
(c) Vier dauerhafte Strukturen



(d) Zwei dauerhafte Strukturen

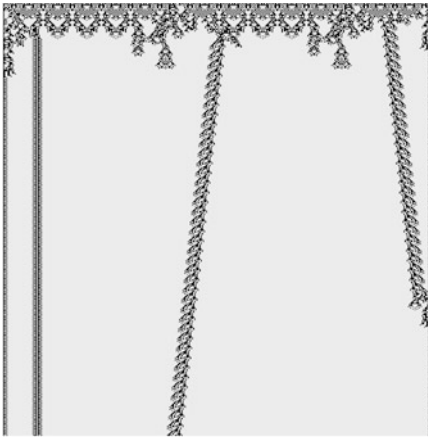


(e) Eine dauerhafte, bewegliche Struktur

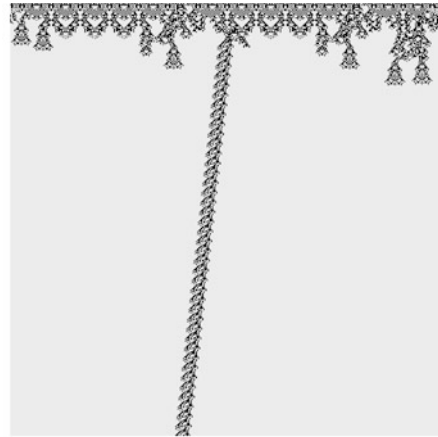


(f) Zwei sich auslöschende Strukturen

Abb. 1.9 Beispiele zum eindimensionalen Universum auf einem 400 Pixel großen Spielfeld mit unterschiedlichen und zufälligen Startkonfigurationen. Zu sehen sind jeweils die ersten 400 Generationen, wobei die Startkonfiguration jeweils der obersten Zeile entspricht. In den Beispielen (a) und (b) stirbt die Population aus, jedoch unterschiedlich schnell. In (c) bis (e) erhalten wir Strukturen, die dauerhaft bestehen bleiben. Zu bemerken ist, dass sich im Beispiel (e) eine bewegliche Struktur gebildet hat. In (f) haben wir den Sonderfall, dass sich zwei Strukturen gegenseitig auslösen, obwohl beide Strukturen für sich alleine betrachtet dauerhaft überlebt hätten



(a) Abgeschlossene Randbedingungen



(b) Periodische Randbedingungen

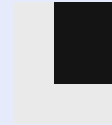
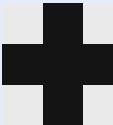
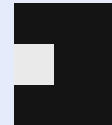
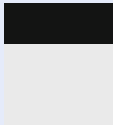
Abb. 1.10 Beispiel zum Einfluss von Randbedingungen beim eindimensionalen Universum. Zur exakt identischen Startgeneration in der jeweils obersten Zeile im Bild wurde die Simulation einmal mit (a) abgeschlossenen Randbedingungen und einmal mit (b) periodischen Randbedingungen durchgeführt

1.5 Game of Life

Neben dem eindimensionalen Universum möchten wir in diesem Abschnitt noch einen zweidimensionalen ¹  zellulären Automaten vorstellen, das **Game of Life** oder **Spiel des Lebens**. Wie der Name schon sagt, handelt es sich dabei wieder um eine Simulation zur Entwicklung des Lebens. Vorgeschlagen wurde das Modell von Conway und veröffentlicht in Gardner (1970). Mit der Zeit wurden neben unzähligen Varianten und Verallgemeinerungen auch theoretische Ergebnisse zum Game of Life präsentiert, wie sie z. B. in Adamatzky (2010) zusammengefasst wurden. Wir beschränken uns in diesem Abschnitt aber auf das einfachste und grundlegende Modell.

Wieder besteht die Zustandsmenge aus den zwei Zuständen ²  der Tab. 1.1, also Pixel besetzt und Pixel nicht besetzt. Als Nachbarschaft wählen wir die Moore-Nachbarschaft ³  aus Abb. 1.4b, welche genau neun Pixel enthält. Weiterhin untersuchen wir periodische Randbedingungen ⁵ .

Tab. 1.4 Beispiele der Regeln im Game of Life. Dargestellt sind jeweils drei mögliche Nachbarschaften von x_{ij} zum Zeitpunkt t_k , die im folgenden Schritt t_{k+1} zu einem leeren oder besetzten Pixel x_{ij} führen. Dabei ist x_{ij} zum Zeitpunkt t_k jeweils das mittlere Pixel

Nachbarschaft von x_{ij} zum Zeitpunkt t_k					x_{ij} zum Zeitpunkt t_{k+1}
	oder		oder		→ 
	oder		oder		→ 
	oder		oder		→ 
	oder		oder		→ 
	oder		oder		→ 

Die Regeln 4 $\circ\circ$ zur Definition eines Schrittes lassen sich durch folgende Fallunterscheidung beschreiben:

1. Wenn x_{ij} besetzt ist und in der Nachbarschaft von x_{ij} (inklusive x_{ij} selbst) weniger als drei der neun möglichen Pixel besetzt sind, dann stirbt auch x_{ij} aufgrund einer Unterbevölkerung aus, sodass x_{ij} in der folgenden Generation nicht besetzt ist, s. Zeile 1 in Tab. 1.4.
2. Wenn x_{ij} besetzt ist und in der Nachbarschaft von x_{ij} (inklusive x_{ij} selbst) drei oder vier der neun möglichen Pixel besetzt sind, dann überlebt x_{ij} , sodass x_{ij} auch in der folgenden Generation besetzt ist, s. Zeile 2 in Tab. 1.4.
3. Wenn x_{ij} besetzt ist und in der Nachbarschaft von x_{ij} (inklusive x_{ij} selbst) mehr als vier der neun möglichen Pixel besetzt sind, dann stirbt auch x_{ij} aufgrund einer Überbevölkerung aus, sodass x_{ij} in der folgenden Generation nicht besetzt ist, s. Zeile 3 in Tab. 1.4.

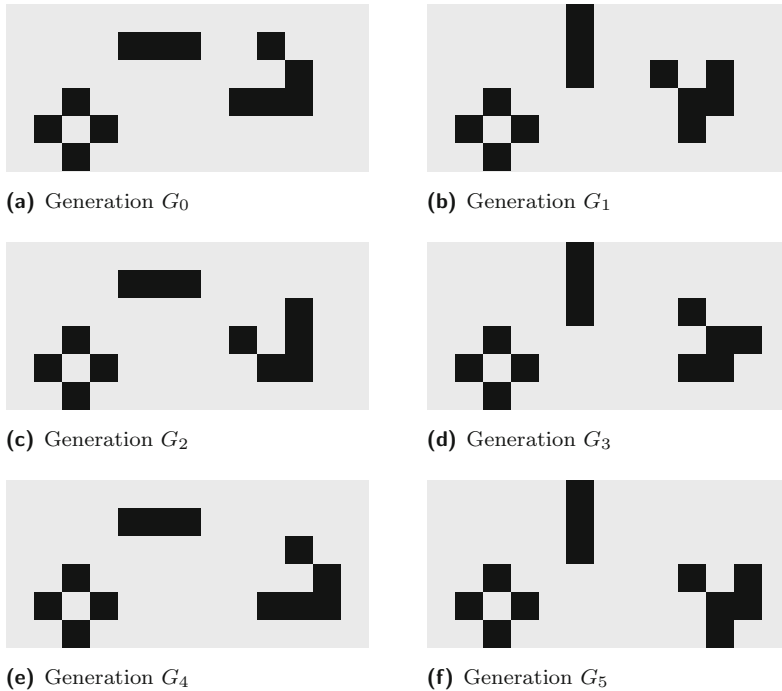


Abb. 1.11 Veranschaulichung der Regeln im Game of Life auf einem zweidimensionalen Spielfeld mit 6×13 Pixel. Gezeigt sind die Startkonfiguration und die ersten fünf Generationen, die sich durch Anwendung der Regeln ergeben

4. Wenn x_{ij} nicht besetzt ist und in der Nachbarschaft von x_{ij} genau drei der neun möglichen Pixel besetzt sind, dann wird ein neues Individuum reproduziert, sodass x_{ij} in der folgenden Generation besetzt ist, s. Zeile 4 in Tab. 1.4.
5. Wenn x_{ij} nicht besetzt ist und in der Nachbarschaft von x_{ij} mehr oder weniger als drei der neun möglichen Pixel besetzt sind, dann passiert nichts, sodass x_{ij} auch in der folgenden Generation nicht besetzt ist, s. Zeile 5 in Tab. 1.4.

Es sei bemerkt, dass für alle Pixel stets genau einer dieser fünf Fälle auftritt, sodass die Schritte wohldefiniert sind.

Zur weiteren Veranschaulichung der Regeln kann in Abb. 1.11 ein Beispiel auf einem 6×13 Pixel großen Spielfeld entnommen werden. In Abb. 1.11a ist die Startkonfiguration G_0 , also Generation G_0 , zu finden. Nach Durchführung der ersten Schritte werden die folgenden fünf Generationen gezeigt.

Wie wir Abb. 1.11 zudem entnehmen können, bilden sich Strukturen mit unterschiedlichen Eigenschaften:

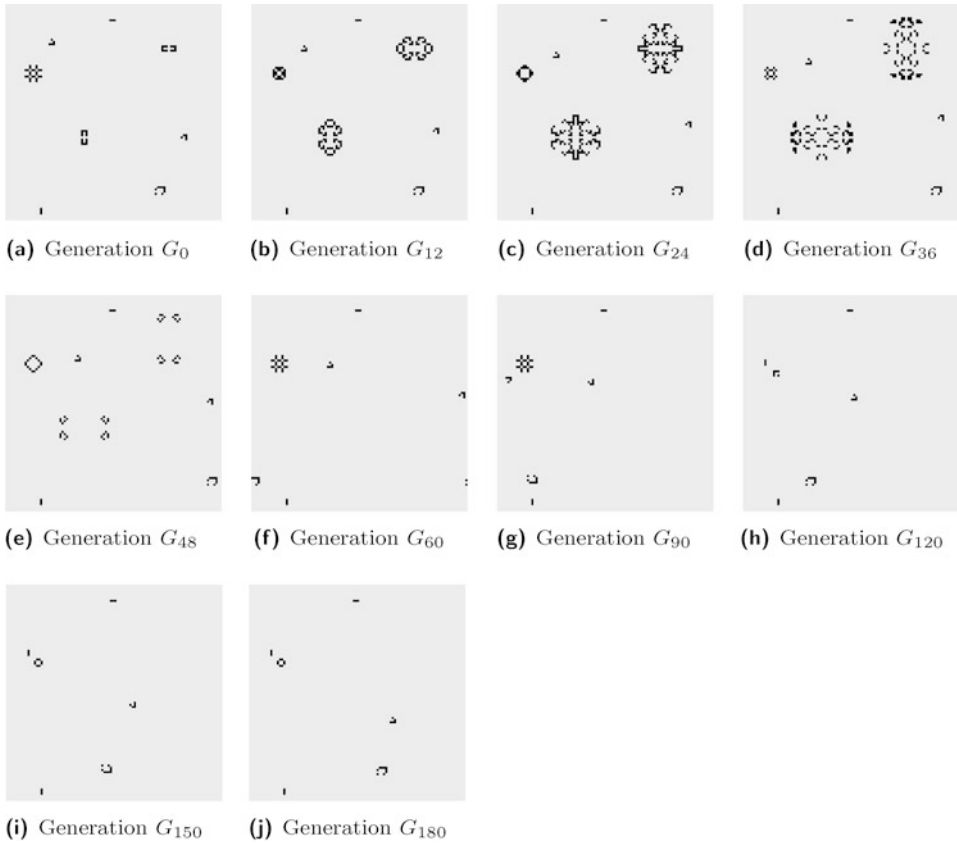


Abb. 1.12 Beispiel zum Game of Life auf einem zweidimensionalen Spielfeld mit 100×100 Pixeln und einer künstlich erzeugten Startkonfiguration, s. (a). Es ergeben sich unterschiedliche Strukturen, von denen einige zunächst stark wachsen, später allerdings aussterben, s. (b) bis (j). Andere Strukturen sind statisch, oszillierend oder dynamisch

1. Es entstehen **statische Strukturen**, die von Generation zu Generation unverändert bleiben. Eine solche Struktur ist im Spielfeld in Abb. 1.11 in der linken Spielfeldhälfte zu finden.
2. Zudem gibt es **oszillierende Strukturen**, s. mittlere Spielfeldhälfte in Abb. 1.11. Im Beispiel handelt es sich um eine Struktur, die alle zwei Generationen identisch ist. Es sind auch andere oszillierende Strukturen möglich, die sich jeweils nach einer größeren Anzahl von Schritten wiederholen.
3. Weiterhin entstehen **dynamische Strukturen** oder **Gleiter**. Auch dies sind Strukturen, die sich nach einer bestimmten Anzahl von Schritten wiederholen, allerdings nicht an der gleichen Stelle wie zuvor. Ein Beispiel eines Gleiters bestehend aus fünf Pixeln mit einer **Periode** von vier Generationen ist in der rechten Spielfeldhälfte zu finden. Die

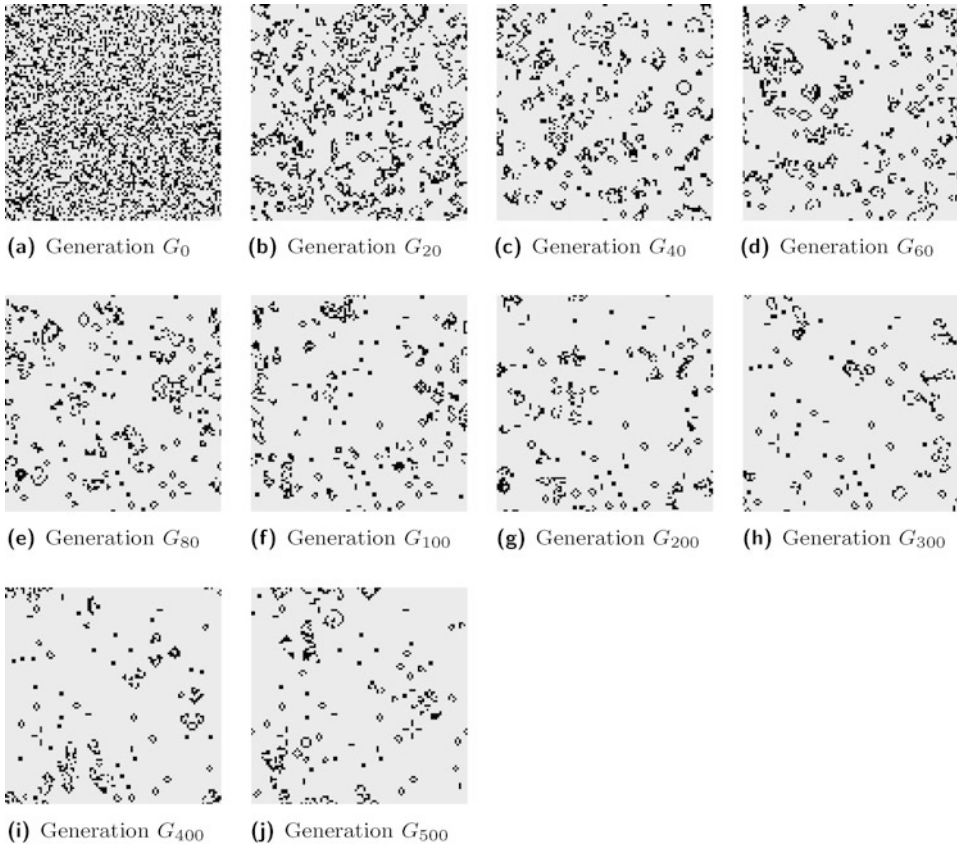


Abb. 1.13 Beispiel zum Game of Life auf einem zweidimensionalen Spielfeld mit 100×100 Pixeln. In (a) wurde eine zufällige Startkonfiguration erzeugt, bei der jedes Pixel mit einer Wahrscheinlichkeit von $\frac{1}{3}$ besetzt und dementsprechend mit einer Wahrscheinlichkeit von $\frac{2}{3}$ nicht besetzt ist. Einige folgende Generationen sind in (b) bis (j) dargestellt

Struktur in Abb. 1.11b ist z. B. identisch mit der Struktur in Abb. 1.11f, allerdings hat sich die gesamte Struktur im Spielfeld verschoben.

Als weiteres Beispiel zum Game of Life haben wir auf einem zweidimensionalen Spielfeld mit 100×100 Pixeln die Startkonfiguration **6** aus Abb. 1.12a gewählt. Es ist zu erkennen, dass sich neben den zuvor genannten Strukturen auch weitere Gebilde entwickeln, die zunächst wachsen, später allerdings aussterben.



Ein letztes Beispiel ist Abb. 1.13 zu entnehmen. Die Startkonfiguration **6** wurde hier zufällig generiert, sodass jedes Pixel mit einer Wahrscheinlichkeit von $\frac{1}{3}$ besetzt ist. Zu beobachten ist, dass die Anzahl der Individuen, also die Anzahl der besetzten Pixel, zunächst abnimmt, zu späteren Zeitpunkten allerdings stabil zu sein scheint.

Pixelspiele

Modellieren und Simulieren mit zellulären Automaten

Scholz, D.

2014, XI, 136 S. 77 Abb. in Farbe., Softcover

ISBN: 978-3-642-45130-0