
13 ZA-Modelle mit kleinen Zustandsmengen

Im vorangegangenen Kapitel wurde das Konzept der Zellularautomaten auf eine Art und Weise benutzt, die bei manchen Leuten eine gewisse Kritik hervorruft. Zwar ist die Zustandsmenge endlich, aber bei den in jeder Zelle gespeicherten Werten handelte es sich um eine oder gar mehrere Fließkommazahlen, die man für „vernünftige“ Ergebnisse üblicherweise mit 32 Bits oder mehr abspeichern muss. Dies bedeutet immerhin 2^{32} Zustände und bei zweidimensionaler von-Neumann-Nachbarschaft 2^{160} verschiedene Argumenttupel für die lokale Überföhrungs-funktion.

In diesem Kapitel wollen wir – zum Teil für die gleichen „Systeme“ wie im letzten Kapitel – Zellularautomatenmodelle angeben, bei denen bewusst darauf geachtet wird, dass die Zustandsmenge vergleichsweise klein ist. Wir beginnen mit einem Zellularautomaten mit einem einzigen Bit Speicher pro Zelle.

Für die späteren Beispiele wird es hilfreich sein, in den Zellen quasi-unabhängige Pseudozufallsbits zur Verfügung zu haben. Bits, die zumindest näherungsweise diesen Anforderungen genügen, kann man auf erstaunlich einfache Art und Weise erzeugen. (Es gibt aber auch Fälle, in denen man mehr Aufwand für die Erzeugung ausreichend guter Zufallsbits treiben muss.)

- 13.1 BEISPIEL. Wir betrachten zweidimensionale Zellularautomaten mit von-Neumann-Nachbarschaft. Zur Abkürzung sei $C = (0,0)$ und N, O, S und W mögen die Nachbarn in den vier „Himmelsrichtungen“ bezeichnen.

Umfasst der Zustand jeder Zelle ein Bit, so führt die Anwendung der lokalen Überföhrungs-funktion

$$\delta(\ell) = \ell(C) \cdot \ell(O) \oplus \ell(N) \oplus \ell(S) \oplus \ell(W)$$

für praktisch jede Anfangskonfiguration, in der nicht alle Zellen im Zustand 0 sind, nach einigen hundert Schritten zu Konfigurationen, in denen die 1-Bits quasi zufällig verteilt sind.

- 13.2 BEISPIEL. Möchte man zwei Pseudozufallsbits erzeugen, so ist zum Beispiel die folgende Vorgehensweise sinnvoll. Um zu erfahren, warum das so ist, konsultiere man etwa das Kapitel über „Random numbers“ in dem Buch von Knuth 1981. Der Zustand q einer Zelle bestehe aus zwei Bits $q[0]$ und $q[1]$:

$$\delta(\ell)[0] = \ell(C)[0] \cdot \ell(O)[0] \oplus \ell(N)[0] \oplus \ell(S)[0] \oplus \ell(W)[1]$$

$$\delta(\ell)[1] = \ell(C)[1] \cdot \ell(O)[1] \oplus \ell(N)[1] \oplus \ell(S)[1] \oplus \ell(W)[0]$$

Ein einfaches aber wichtiges Beispiel, bei dem man Pseudozufallsbits benötigt, sind *Random Walks* (im Deutschen manchmal auch als Irrfahrten bezeichnet).

- 13.3 Die Aufgabe, die es mit einem Zellularautomaten zu realisieren gilt, ist im Eindimensionalen dabei die folgende: Es gibt ein „Objekt“, den „Random Walker“, der sich in diskreten Schritten bewegen kann (man stelle sich etwa vor, auf den Felder eines Turingmaschinenbandes), und war in jedem Schritt zufällig mit einer gewissen festen Wahrscheinlichkeit p_n vom momentan besuchten Feld auf ein Nachbarfeld n . Sinnvollerweise verlangt man $\sum_{n \in \mathbb{N}} p_n = 1$.

Drei einfache Beispiele für entsprechende Random Walks sind:

- $p_1 = p_{-1} = 1/4$ und $p_0 = 1/2$: Dieser Random Walker bleibt manchmal stehen.

- $p_1 = p_{-1} = 1/2$: Dieser Random Walker bewegt sich in jedem Schritt.
- $p_{-2} = 1/9, p_{-1} = 2/9, p_0 = 2/9, p_1 = 4/9$: Dieser Random Walker hat Gleichgewichtsstörungen.

13.4 RECHNUNG. Wir wollen plausibel machen, dass eine sich durch Random Walks ergebende Dynamik jedenfalls manchmal näherungsweise der Diffusionsgleichung genügt. Der Einfachheit halber betrachten wir den eindimensionalen Fall mit $H_1^{(1)}$ -Nachbarschaft und bezeichnen mit $p(t, x)$ die Wahrscheinlichkeit, dass sich zum Zeitpunkt t an der Stelle x ein Partikel befindet. Ein Schritt der Überföhrungsfunktion entspreche einem Fortschreiten der Zeit um Δt und Nachbarzellen seien in beiden Richtungen Δx entfernt. Ein Partikel bewege sich in jedem Schritt mit gleicher Wahrscheinlichkeit $1/2$ in eine der beiden Richtungen. Dann gilt:

$$p(t + \Delta t, x) = \frac{1}{2}p(t, x - \Delta x) + \frac{1}{2}p(t, x + \Delta x) \quad (13.1)$$

Taylorentwicklung der einzelnen Summanden und (geeignetes) Abbrechen nach den ersten Summanden ergibt (siehe 12.4):

$$\begin{aligned} p(t + \Delta t, x) &\approx p(t, x) + \Delta t \frac{\partial p}{\partial t}(t, x) \\ p(t, x - \Delta x) &\approx p(t, x) - \Delta x \frac{\partial p}{\partial x}(t, x) + \frac{(\Delta x)^2}{2} \frac{\partial^2 p}{\partial x^2}(t, x) \\ p(t, x + \Delta x) &\approx p(t, x) + \Delta x \frac{\partial p}{\partial x}(t, x) + \frac{(\Delta x)^2}{2} \frac{\partial^2 p}{\partial x^2}(t, x) \end{aligned}$$

Setzt man dies in Gleichung 13.1 ein, so erhalt man

$$\begin{aligned} p(t, x) + \Delta t \frac{\partial p}{\partial t}(t, x) &\approx \frac{1}{2}p(t, x) - \frac{1}{2}\Delta x \frac{\partial p}{\partial x}(t, x) + \frac{1}{2} \frac{(\Delta x)^2}{2} \frac{\partial^2 p}{\partial x^2}(t, x) \\ &\quad + \frac{1}{2}p(t, x) + \frac{1}{2}\Delta x \frac{\partial p}{\partial x}(t, x) + \frac{1}{2} \frac{(\Delta x)^2}{2} \frac{\partial^2 p}{\partial x^2}(t, x) \end{aligned}$$

also

$$\frac{\partial p}{\partial t}(t, x) \approx \frac{(\Delta x)^2}{2\Delta t} \frac{\partial^2 p}{\partial x^2}(t, x)$$

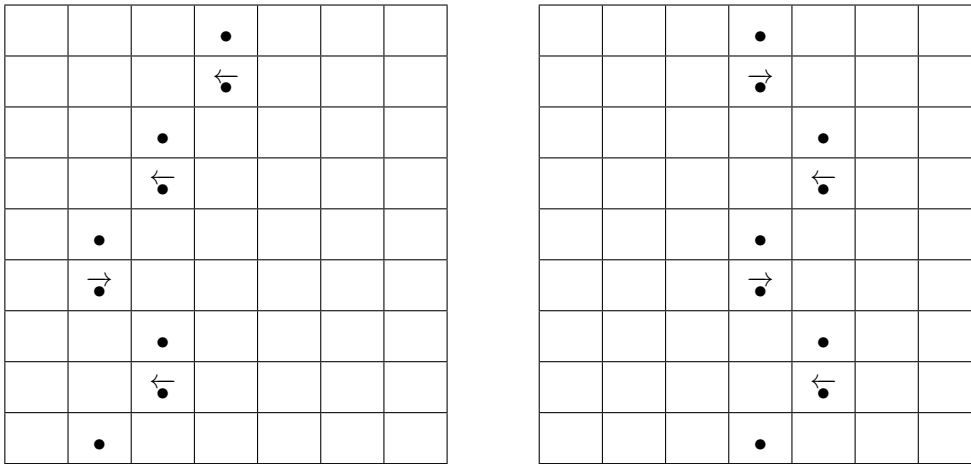
Wenn es gelingt, Random Walks in Zellularautomaten zu realisieren, dann haben wir damit also eine weitere Methode, um Diffusion zu modellieren. Wir bleiben zunachst bei dem Fall $p_1 = p_{-1} = 1/2$.

13.5 Fur die Zufallskomponente kann man entweder auf Pseudozufallsbits wie in den Beispielen 13.1 und 13.2 zuruckgreifen oder stochastische Zellularautomaten (siehe Definition 12.20) benutzen.

Wenn man mochte, dass aus einem Random Walker nach einer Zufallsbewegung nicht 2 oder 0 Random Walker werden, dann muss (machen Sie sich klar, warum!) man einen Zufallsschritt durch zwei Zellularautomatenschritte realisieren. Die Zelle, die gerade vom Random Walker besucht wird, ist z. B. durch eine Markierung \bullet gekennzeichnet.

- Im ersten Schritt wurde die markierte Zelle gleichwahrscheinlich in einen der Zustande $\leftarrow \bullet$ oder $\bullet \rightarrow$ ubergehen.
- Im zweiten kann diese Zelle die Markierung vergessen und je nach Richtungsinformation die eine oder die andere Nachbarzelle die Markierung speichern.

Zwei mogliche legale Berechnungen sind dann etwa:



13.6 Aber was, wenn man nun *zwei* Random Walker haben möchte? Wenn man den eben vorgestellten Ansatz weiterverfolgen will, muss man festlegen, in welchen Zustand eine Zelle übergeht, die bei beiden Nachbarn eine Markierung beobachtet, die sich zu ihr bewegen „will“.

Sollen beide Random Walker *völlig unabhängig* voneinander sein, dann muss man die Zustandsmenge so erweitern, dass eine Zelle auch zwei Markierungen aufnehmen kann; und δ muss es zulassen, dass sich beide im nächsten Schritt zum gleichen Nachbarn bewegen.

Diese Vorgehensweise hilft aber nicht weiter, wenn man auch den Fall erfassen will, dass in der Anfangskonfiguration eine beliebig große Anzahl von Markierungen vorhanden ist. Dann kann man die Random Walker nicht mehr unabhängig voneinander laufen lassen, sondern muss z. B. sicherstellen, dass nie in einer Zelle mehr als einer oder jedenfalls mehr als eine konstante Anzahl von ihnen ist.

Das ist ein typisches Beispiel für ein *Ausschlussprinzip*. Sie kann man auf verschiedene Weise realisieren. Eine Möglichkeit ist, bei einem „normalen“ Zellularautomaten die lokale Überföhrungsfunktion geeignet zu definieren. Tun Sie dies zur Übung, um festzustellen, wie umständlich das ist (man braucht zusätzliche Zustände).

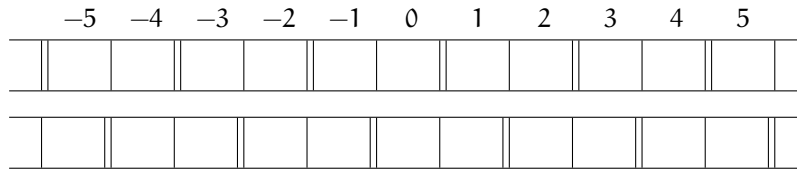
Es gibt aber elegantere Methoden, die auf zwei leicht modifizierten und miteinander verwandten Definitionen von Zellularautomaten beruhen. Sie wollen wir im folgenden am Beispiel der Random Walks vorstellen.

13.7 DEFINITION (LOKALE BLOCKÜBERFÖHRUNGSFUNKTION) Eine deterministische *lokale Blocküberföhrungsfunktion* β legt für jede lokale Konfiguration $\ell : \mathbb{N} \rightarrow Q$ neue Zustände für *alle* Zellen in der Nachbarschaft fest, also: $\beta : Q^{\mathbb{N}} \rightarrow Q^{\mathbb{N}}$.

Eine *stochastische* lokale Blocküberföhrungsfunktion β legt für jede lokale Konfiguration $\ell : \mathbb{N} \rightarrow Q$ eine Wahrscheinlichkeitsverteilung für neue Zustände aller Zellen in der Nachbarschaft fest, also: $\beta : Q^{\mathbb{N}} \rightarrow [0; 1]^{Q^{\mathbb{N}}}$ (mit der üblichen Forderung, dass die Summe der Wahrscheinlichkeiten jeweils 1 ergeben muss).

Für die Anwendung einer lokalen Blocküberföhrungsfunktion in einer Konfiguration wird der gesamte Raum R in Kacheln der durch N beschriebenen Form partitioniert (d. h. der Raum wird vollständig durch disjunkte Kacheln überdeckt) und dann auf jeder Kachel β angewendet. Wir sprechen in diesem Fall auch von einem *Block-Zellularautomaten*. \diamond

13.8 BEISPIEL. Betrachten wir den eindimensionalen Fall und $N = \{0, 1\}$. Eine Kachel besteht also aus zwei nebeneinanderliegenden Zellen. Für eine vollständige Kachelung aller Zellen gibt es zwei Möglichkeiten; Zelle 0 ist entweder linker oder rechter Partner in seiner Kachel. Die Kachelgrenzen sind in der folgenden Abbildung durch doppelte senkrechte Striche dargestellt.



- 13.9 Damit bei wiederholter Anwendung einer Blocküberföhrungsfunktion Informationsaustausch auch über größere Entfernungen und nicht nur immer wieder innerhalb der gleichen Kachel möglich ist, muss man in aufeinanderfolgenden Schritten offensichtlich verschiedene Kachelungen des Raumes benutzen.

Damit das ganze Vorgehen offensichtlich berechenbar bleibt und man auch nicht „durch ein Hintertürchen versteckt“ zusätzliche Informationen zur Verfügung stellt, wählt man üblicherweise eine Reihe von Kachelungen, die dann immer zyklisch benutzt wird.

Je nachdem, welche Dimension der Raum und welche Form die Kacheln haben, werden die Details aber natürlich verschieden sein.

- 13.10 BEISPIEL. Für den schon eben in Beispiel 13.8 Fall wird man die beiden oben dargestellte Kachelungen abwechselnd benutzen.

Wer sich an den Odd-Even-Transposition-Sort erinnert, dem wird das hoffentlich nicht gänzlich unbekannt vorkommen. Ihn hätte man bequem als Blockzellularautomat darstellen können, indem man Kachelform $N = \{0, 1\}$ wählt und als Blocküberföhrungsfunktion:

$$\beta([a, b]) = \begin{cases} [a, b] & \text{falls } a \leq b \vee a = \# \vee b = \# \\ [b, a] & \text{falls } a > b \end{cases}$$

Im Zweidimensionalen am häufigsten anzutreffen ist die folgende Vorgehensweise:

- 13.11 DEFINITION Die *Margolus-Nachbarschaft* im Zweidimensionalen ist dadurch gekennzeichnet, dass man eine lokale Blocküberföhrungsfunktion benutzt und immer abwechselnd zwei Kachelungen von \mathbb{Z}^2 in 2×2 -Blöcke zugrunde legt, bei denen sich jede Zelle abwechselnd in einer und in der diametral gegenüberliegenden Ecke der Kachel befindet.

- 13.12 LEMMA. Jeder Block-Zellularautomat mit einer zyklisch durchlaufenen Folge von Kachelungen kann von einem normalen Zellularautomaten (mit gegebenenfalls größerer Zustandsmenge und Nachbarschaft) simuliert werden.

- 13.13 ÜBUNG. Beweisen Sie dieses Lemma. Überlegen Sie sich, ob auch die Umkehrung dieses Lemmas gilt und beweisen Sie sie gegebenenfalls.

Damit haben wir das nötige Handwerkszeug, um nun einen Zellularautomaten mit nur 4 Zuständen angeben zu können, dessen Verhalten wieder mit Diffusion zu tun hat.

- 13.14 BEISPIEL. Der Zustand b jeder Zelle beinhaltet ein Bit. Dieses Bit ist eine „Stelle“, an der sich ein „Partikel“ befinden kann ($b = 1$) oder die frei sein kann ($b = 0$). Es werde die Margolus-Nachbarschaft verwendet.

Die Blocküberföhrungsfunktion ist probabilistisch: Zufällig gleichwahrscheinlich werden in jedem Schritt die vier Partikelbits der Kachel im Uhrzeigersinn oder entgegen dem Uhrzeigersinn gedreht.

Wendet man diese Überföhrungsfunktion auf eine Konfiguration an, in der genau eine Zelle ein Partikel enthält, so kann man einen Random Walk dieses Teilchens beobachten.

Hat man in der Anfangskonfiguration mehrere Teilchen, so ist durch ihr gleichzeitiges Verschieben innerhalb einer Kachel immer sichergestellt, dass sich nicht nach und nach ganz viele Teilchen in der gleichen Zelle einfinden. In der Simulation beobachtet man diffusionsähnliches Verhalten.

Auch für manche komplizierteren Phänomene aus der Realität liefern Zellularautomaten, bei denen sich „Partikel“ geeignet durch den Raum bewegen, gute Modellierungen.

13.15 *Gittergase* (engl. *lattice gases*) sind Modelle für (manche) strömende Flüssigkeiten und Gase. Auch hier benutzt man die Modellierung mittels einzelner Partikel.

Im einfachsten Fall ist in jeder Zelle für jeden Nachbarn ein Bit vorgesehen. Eine 1 bedeutet, dass ein Partikel vorhanden ist, das sich im nächsten Schritt zum entsprechenden Nachbarn bewegt. Diese Bewegungen der Partikel bilden die erste Phase der Überföhrungsfunktion. In einer zweiten werden üblicherweise noch „Kollisionen“ von aus verschiedenen Richtungen in einer Zelle eintreffenden Partikeln behandelt.

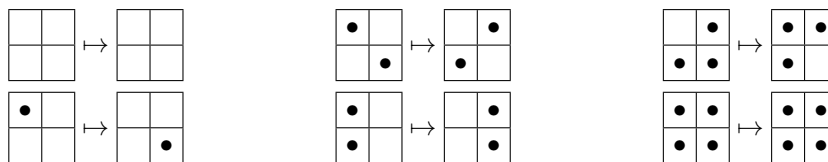
Dabei sind die Überföhrungsfunktionen häufig so gewählt, dass

- Rotationssymmetrie vorliegt,
- „Massenerhaltung“ gewährleistet ist, d. h. sich die Anzahl der Partikel beim Übergang von einer Konfiguration zur nächsten nicht ändert sowie häufig so, dass
- „Impulserhaltung“ gegeben ist. Darunter ist zu verstehen, dass die Summe aller Vektoren, die man erhält, wenn man für jedes Partikel die Differenz von Ausgangs- und Zielzelle in einem Schritt bildet, Null ist.

In manchen Fällen diese Eigenschaften mikroskopisch exakt ablesbar, in anderen erscheint es zumindest plausibel, dass sie zumindest im Großen gegeben sind.

13.16 BEISPIEL. (HPP-GAS) Bei HPP-GAS benutzt man im Zweidimensionalen die $H_1^{(2)}$ -Nachbarschaft, so dass in jeder Zelle vier Bits (für die maximal vier Partikel zu den vier Nachbarn) nötig sind. Als Kollisionen wird nur angesehen, wenn genau zwei Partikel zusammentreffen, die aus entgegengesetzten Richtungen kommen müssen. Sie werden dann um 90° abgelenkt.

Eine einfache Variante von HPP-GAS lässt sich bequem mit Hilfe eines Block-ZA beschreiben. Die Kollisionphase wird durch die Blocküberföhrungsfunktion realisiert, die Bewegungsphase durch den Wechsel der Partitionierung. Die Blocküberföhrungsfunktion ist im wesentlichen durch die folgende Abbildung beschrieben:



Für Blöcke, die nicht aufgeföhrt sind, ist aber jedenfalls eine rotierte Variante davon vorhanden; die gleiche Rotation hat man dann auch beim Bild vorzunehmen, um das gesuchte Bild zu erhalten.

Eine mit Blockzellularautomaten verwandte Alternative um z. B. auch Ausschlussprinzipien zu realisieren sind sogenannte partitionierte Zellularautomaten.

13.17 DEFINITION Ein *partitionierter Zellularautomat* C ist dadurch charakterisiert, dass die Zustandsmenge von der Form $Q = P^N$ ist für eine endliche Menge P und dass die Arbeitsweise von C beschrieben werden kann durch eine Abbildung $\delta' : P^N \rightarrow P^N$.

Bei einem partitionierten Zellularautomaten wird in einer Konfiguration c für die Bestimmung des neuen Zustandes einer Zelle i von der Zelle an der Stelle $i + n$ nicht deren ganzer Zustand verwendet, sondern nur die Zustandskomponente $c(i + n)(n)$. \diamond

- 13.18 Eine Randbemerkung zu reversiblen Zellularautomaten: Das „schöne“ an partitionierten Zellularautomaten ist, dass man sofort erkennen kann, ob die globale Überföhrungsfunktion injektiv ist. Das ist nämlich genau dann der Fall, wenn δ' injektiv ist. Überlegen Sie sich als Übung, dass die zuletzt gemachte Aussage für partitionierte Zellularautomaten richtig ist, gleich welcher Dimension. Für „klassische“ Zellularautomaten im Zwei- und Höherdimensionalen ist die globale Injektivität *unentscheidbar!*

Auch bei Gittergasen ist es mitunter vorteilhaft (i.e. die Simulationsergebnisse können besser bzw. leichter mit der Realität in übereinstimmung gebracht werden), (pseudo-)probabilistische Überföhrungsfunktionen zu verwenden. Im folgenden stellen wir einen eine partitionierten solchen vor.

- 13.19 BEISPIEL. FHP-GAS benutzt das in Abbildung 13.1 dargestellte *hexagonale Gitter*, bei dem jede Zelle sechs Nachbarn hat. Man mache sich klar, dass man ein unendliches hexagonales Gitter

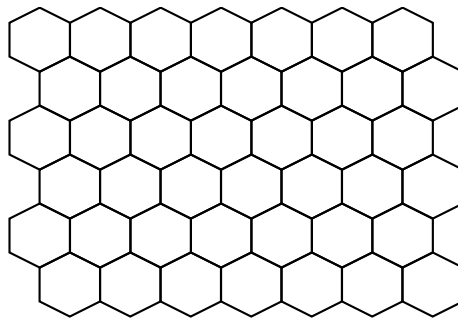


Abbildung 13.1: Hexagonales Gitter.

leicht auf einem orthogonalen Gitter mit $M_1^{(2)}$ -Nachbarschaft simulieren kann. Schwieriger wird es allerdings im endlichen, zyklisch geschlossenen Fall.

Die in der Originalarbeit von Frisch, Hasslacher und Pomeau 1986 angegebene Kollisionsregel lautet: Wenn der „Gesamtimpuls“ der ankommenden Teilchen 0 ist und nicht 4 Teilchen ankommen, dann werden alle Teilchen um 60° im Uhrzeigersinn oder entgegengesetzt dem Uhrzeigersinn abgelenkt. Eine Variante fordert diese Kollisionsstrategie auch für das Zusammentreffen von vier Teilchen.

Für die Wahl der Rotationsrichtung gibt es verschiedene Möglichkeiten:

- In jeder Konfiguration wird der Drehsinn für alle Zellen einheitlich und immer abwechselnd gewählt. ((mikroskopisch) deterministisch)
- In jeder Konfiguration wird der Drehsinn für alle Zellen einheitlich zufällig gewählt.
- In jeder Zelle wird bei jeder Kollision der Drehsinn zufällig gewählt.

Eine noch bessere Variante ist FHP III. Hier gibt es neben den sechs Plätzen für Partikel mit nicht verschwindendem Impuls auch noch Platz für ein ruhendes Teilchen. Es ist in der Abbildung 13.2, die die Überföhrungsfunktion beschreibt, durch ein \bullet dargestellt. Sich bewegende

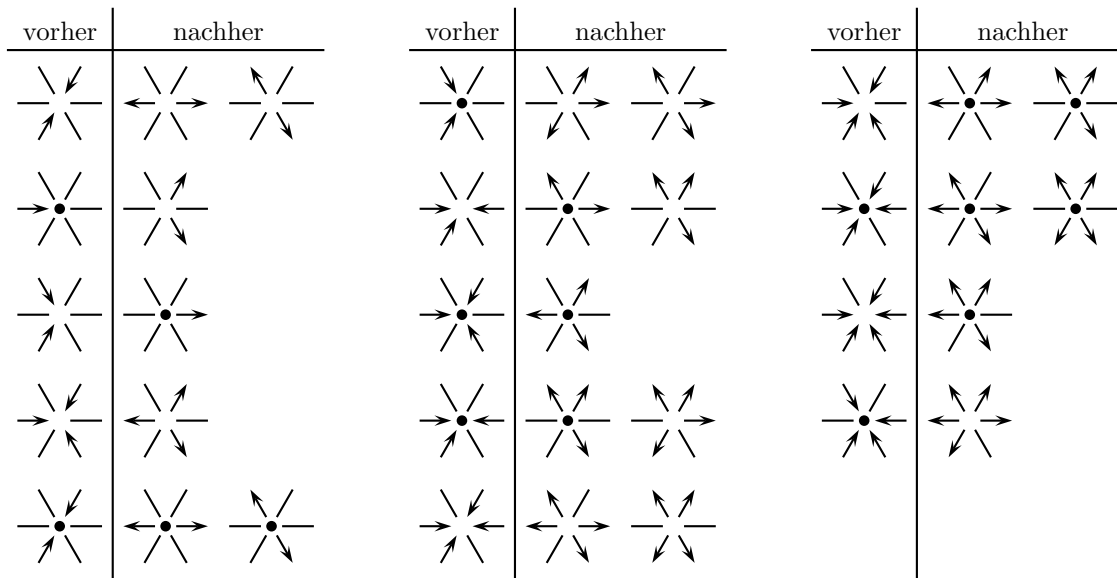


Abbildung 13.2: Die Kollisionsregeln für FHP III. Falls in einer Situation zwei Möglichkeiten angegeben sind, werden sie mit gleicher Wahrscheinlichkeit gewählt.

Partikel sind durch Pfeile dargestellt und freie Plätze durch dünne Striche. Nicht dargestellte Situationen ergeben sich durch Rotation aus angegebenen.

Der Vorteil von FHP III gegenüber HPP besteht darin, dass asymptotisch letzteres die Navier-Stokes Differentialgleichung nur näherungsweise erfüllt, ersteres aber exakt. Leider benötigt man z. B. für aussagekräftige Bilder große Zellularautomaten (mehrere Millionen Zellen), die über viele zehntausend Schritte simuliert werden müssen.

Das Buch von Rothman und Zaleski (1997) bietet eine umfassende Einführung in die Simulation hydrodynamischer Vorgänge mit Hilfe von Zellularautomaten.

13.20 BEISPIEL. Auch Reaktions-Diffusions-Systeme kann man mit partikelbasierten Ansätzen modellieren. Wir verweisen hier auf die Dissertation von Weimar 1995 und das Buch von Chopard und Droz 1998. Auch das Buch von Weimar (1997) enthält zu diesem Thema weitere Informationen. Es bietet außerdem Informationen über geeignete Verfahren zur *schnellen* Simulation von Zellularautomaten auf Rechnern.

13.21 BEISPIEL. Als nächstes wollen wir kurz auf *Ising Systeme* eingehen (siehe Toffoli und Margolus 1987). Wir betrachten wieder zweidimensionale Zellularautomaten mit $H_1^{(2)}$ -Nachbarschaft. Die Zustandsmenge der Zellen ist zweielementig; die Zustände werden in diesem Zusammenhang auch als *Spins* bezeichnet und \uparrow und \downarrow geschrieben. Dies soll wohl daran erinnern, dass man sich darunter die beiden Orientierungen eines „kleinen“ (Elementar-)Magnetens vorstellen kann. Sind zwei benachbarte Spins einander entgegengesetzt ($\uparrow\downarrow$ oder $\downarrow\uparrow$), so ist „zwischen“ ihnen eine gewisse Menge „Energie“ gespeichert, sind sie dagegen parallel ($\downarrow\downarrow$ oder $\uparrow\uparrow$) dann nicht.

Betrachten wir nun, was passiert, wenn ein Spin seine Orientierung ändern würde. Wenn er vorher die gleiche Richtung hatte wie die Majorität seiner Nachbarn, dann entspräche dies einer Vergrößerung der gespeicherten Energie. Hatte vorher die entgegengesetzte Richtung wie die Majorität seiner Nachbarn, dann entspräche dies einer Verringerung der gespeicherten Energie.

Hatten vorher gleich viele seiner Nachbarn die gleiche bzw. die entgegengesetzte Richtung wie der umkippende Spin, dann bleibt die Gesamtenergie gleich.

Diese Überlegung ist aber nur richtig, solange nur ein Spin, nicht aber auch noch gleichzeitig seine Nachbarn ihre Orientierung ändern.

Für eine Realisierung als Zellularautomat, in dem unter Gewährleistung der Energieerhaltung in jedem Schritt möglichst viele Spins ihre Orientierung ändern sollen, liegt es daher nahe, den Raum „mit einem Schachbrett zu unterlegen“ und immer abwechselnd nur Spins auf weißen bzw. auf schwarzen Feldern zu betrachten und gegebenenfalls gemäß obiger Beschreibung zu ändern.

Dazu kann man ein Bit s in Zellen des Raumes so initialisieren, dass sich ein Schachbrettmuster ergibt. Die lokale Überföhrungsfunktion wird so konstruiert, dass Zellen mit $s = 0$ ihren übrigen Zustand nicht ändern. Zellen mit $s = 1$ tun dies gemäß der obigen Beschreibung: Der Spin wird genau dann gewechselt, wenn genau 2 der 4 Nachbarzellen den gleichen Spin haben wie die Zelle selbst.

Wendet man diese Überföhrungsfunktion auf Anfangskonfigurationen an, bei denen die Spins am Anfang auf verschiedene Art initialisiert sind (Energie, Magnetisierung (Anteil \uparrow minus Anteil \downarrow)) so kann man Entwicklungen beobachten, die bei naiven Erwartungen überraschen aber Zusammenhänge mit physikalischen Phänomene aufweisen.

13.22 BEISPIEL. Abschließend wollen wir noch kurz auf die Modellierung von Verkehrsgeschehen eingehen und den Zellularautomaten von Nagel und Schreckenberg (1992) vorstellen, der trotz seiner Einfachheit den Verkehrsfluss auf einer einspurigen Straße in einer Richtung verblöffend gut nachbildet.

Im Laufe der vergangenen Jahre wurde nicht nur dieses Modell verbessert, sondern zum Beispiel auch auf mehrspurigen Verkehr und auf Straßen mit entgegenkommenden Fahrzeugen verallgemeinert. Außerdem gibt es inzwischen auch Arbeiten, die sich mit der Nachbildung von Straßenkreuzungen und ähnlichem beschäftigen. Im WWW findet man unter <http://www.traffic.uni-duisburg.de/> und unter http://www.tu-dresden.de/vkiwv/vwista/sta1_hp.htm neben anderem auch Java-Applets zur Simulation von Fahrzeug- und Fußgängerströmen.

Üblicherweise diskretisiert man die Straße in Segmente der Länge 7.5 m, denen jeweils eine Zelle entspricht. Eine Zelle kann leer oder von einem Auto besetzt sein. Zur Darstellung seiner Geschwindigkeit v benutzt man eine ganze Zahl zwischen 0 und $v_{\max} = 5$. Nimmt man an, dass ein Schritt des anschließend beschriebenen Zellularautomaten gerade 1 s entspricht, ergibt sich eine modellierte Höchstgeschwindigkeit von

$$1 \frac{\text{Schritt}}{\text{s}} \cdot 5 \frac{\text{Zelle}}{\text{Schritt}} \cdot 7.5 \frac{\text{m}}{\text{Zelle}} = \frac{5 \cdot 7.5 \cdot 3600 \text{ km}}{1000 \text{ h}} = 135 \frac{\text{km}}{\text{h}}$$

Ein Schritt der Überföhrungsfunktion von Nagel und Schreckenberg besteht aus drei Phasen:

1. Beschleunigung: Befinden sich vor einem Fahrzeug genau g freie Zellen, dann „beschleunigt“ es auf $v' \leftarrow \min(v + 1, v_{\max}, g)$.
2. Trödeln: Mit einer kleinen Wahrscheinlichkeit p wird die Geschwindigkeit um 1 erniedrigt: $v'' \leftarrow \max(v' - 1, 0)$.
3. Fahren: Das Fahrzeug bewegt sich um v'' Zellen weiter; $v \leftarrow v''$.

Variationen dieses Ansatzes sind auch z. B. auf das Verhalten von Fußgängern anwendbar.

Zusammenfassung

- In manchen Fällen kommt man auch schon mit wenigen Zuständen pro Zelle zu guten Modellen für reale Phänomene.
- Wichtige Konzepte, die hierbei teils wesentlich oder teils sehr nützlich sein können, sind (Pseudo-)Randomisierung, zB für Random Walks, und Block- und partitionierte Zellularautomaten.

Literatur

- Chopard, Bastien und Michel Droz (1998). *Cellular Automata Modeling of Physical Systems*. Cambridge University Press (siehe S. 105).
- Frisch, Uriel, Brosl Hasslacher und Yves Pomeau (1986). «Lattice-Gas Automata for the Navier-Stokes Equation». In: *Physical Review Letters* 56, S. 1505–1508 (siehe S. 104).
- Knuth, Donald Ervin (1981). *Seminumerical Algorithms*. 2nd. Bd. 2. The Art of Computer Programming. Addison Wesley (siehe S. 99).
- Nagel, Kai und Michael Schreckenberg (1992). «A cellular automaton model for freeway traffic». In: *Journal de Physique I* 2, S. 2221–2229 (siehe S. 106).
- Rothman, Daniel H. und Stéphane Zaleski (1997). *Lattice-Gas Cellular Automata*. Cambridge University Press (siehe S. 105).
- Toffoli, Tommaso und Norman Margolus (1987). *Cellular Automata Machines, A New Environment for Modeling*. Cambridge, MA: MIT Press. ISBN: 0-262-20060-0 (siehe S. 105).
- Weimar, Jörg Richard (1995). «Cellular Automata Methods for Reactive Systems». Diss. Université Libre de Bruxelles, Belgium (siehe S. 105).
- (1997). *Simulation with Cellular Automata*. Berlin: Logos-Verlag (siehe S. 105).