

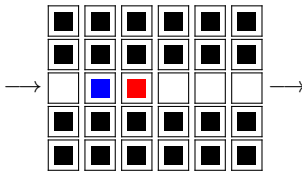
2 Zur Berechnungsmächtigkeit von Zellularautomaten

2.1 ZIELE. In diesem Kapitel wollen wir (unter Benutzung des Beispiels WIREWORLD aus Kapitel 1) zu zwei Aussagen die Beweise skizzieren:

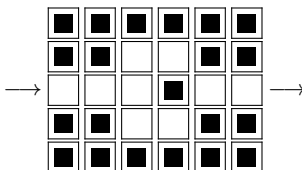
1. Es gibt einen synchron arbeitenden zweidimensionalen deterministischen ZA mit 4 Zuständen und $M_1^{(2)}$ -Nachbarschaft, der einen programmgesteuerten Universalrechner simuliert.
2. Zu jeder Turingmaschine T (kurz: TM) mit einem Band und einem Kopf gibt es einen synchron arbeitenden eindimensionalen deterministischen ZA C mit $H_1^{(1)}$ -Nachbarschaft, der T simuliert.

2.2 BEISPIELE. Der im wesentlichen in 1.5 spezifizierte Zellularautomat heißt WIREWORLD (siehe zum Beispiel Dewdney (1990)). Die folgenden Beispielkonfigurationen stammen aus news:comp.theory.cell-automata vom Februar/März 1990. Im Anfang 2003 wurde in der gleichen Newsgroup eine Verallgemeinerung von WIREWORLD diskutiert, bei der es zwei Sorten von «Elektronenvorderteilen» gibt; siehe z. B. <http://www.geocities.com/fiziwig/cabead1.html>.

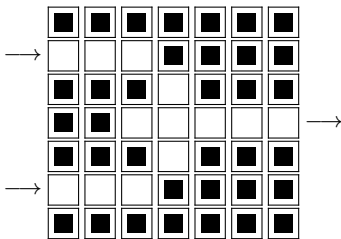
1. Ein «Draht» mit einem «Elektron»:



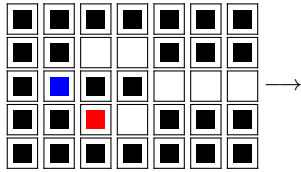
2. Eine «Diode»:



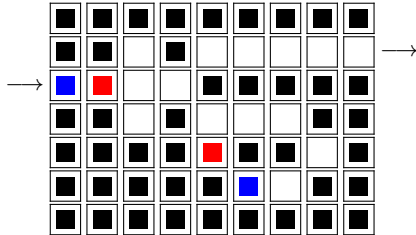
3. Ein «Oder-Gatter»:



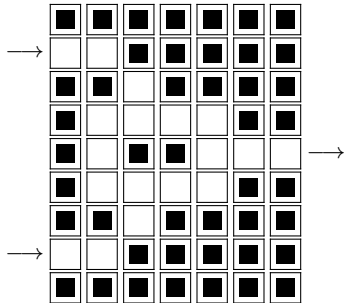
4. Ein «Takterzeuger»:



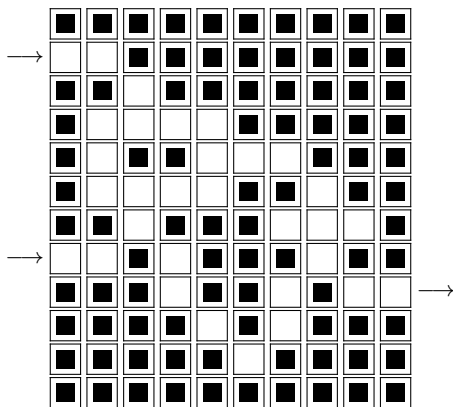
5. Ein «Inverter»:



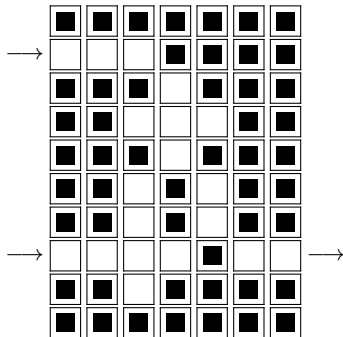
6. Ein «Exklusiv-Oder-Gatter»:



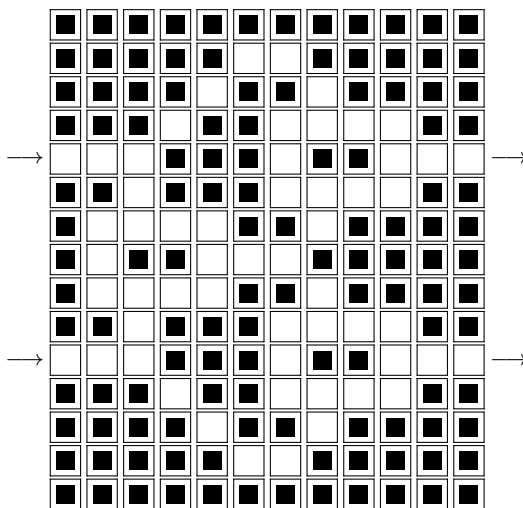
7. Ein «Und-Gatter»:



8. Ein «Speicherelement»; unten ist der Setzeingang und oben der Rücksetzeingang:



9. Eine «Drahtkreuzung» kann man aus drei Exklusiv-Oder-Gattern konstruieren, da $(a \oplus b) \oplus a = b$ ist. (\oplus bezeichne das exklusive Oder.):



2.3 SATZ. Es gibt einen synchron arbeitenden deterministischen zweidimensionalen Zellularautomaten mit $M_1^{(2)}$ -Nachbarschaft und vierelementiger Zustandsmenge, mit dem man jeden endlichen Automaten (mit beliebig großer Zustandsmenge und beliebiger Überföhrungsfunktion) simulieren kann.

2.4 BEWEISSKIZZE. Es werden die folgenden Konventionen vereinbart:

1. Ein «1-Signal» wird durch ein Zeitintervall der Länge 24 repräsentiert, das mit einem Elektron beginnt, auf das in den nächsten 22 Takten nichts folgt.
2. Ein «0-Signal» wird durch ein Zeitintervall der Länge 24 repräsentiert, in dem kein einziges Elektron vorkommt.

Alle in Beispiel 2.2 aufgeföhrten Bausteine können so in Module der Größe 24×24 eingepasst werden, dass gilt: Wird der Eingang bzw. werden gleichzeitig die Eingänge mit 0- oder 1-Signalen versorgt, so liefert der Ausgang 48 Takte später die zugehörige Ausgabe. Dieses «synchronisierte» Vorgehen ist (jedenfalls teilweise) dadurch bedingt, dass die «Inverter» ein Signal nur dann «invertieren», wenn das «Elektron» zum richtigen Zeitpunkt am Eingang ankommt.

Den Rest föhren wir hier nicht aus. ■

2.5 Durch weitere Überlegungen kann man sich davon überzeugen, dass man mit WIREWORLD sogar einen programmgesteuerten Universalrechner simulieren kann. Das zusätzlich auftretende

Problem besteht darin, dass die Größe des für eine Rechnung benötigten Speicherplatzes nicht a priori feststeht und beliebig groß werden kann. Da die Überföhrungsfunktion von WIREWORLD es nicht zulässt, in einer Konfiguration zusätzlichen Draht zu erzeugen, muss man als Anfangskonfiguration eine solche wählen, in der bereits unendlich viel Draht vorhanden ist.

Aufmerksame Leser werden bemerken, dass sich hier ein Abgrund auftut. Denn in einer unendlichen Konfiguration lässt sich natürlich prinzipiell unendlich viel Information kodieren, zum Beispiel die Lösung des Halteproblems für Turingmaschinen. Das will man natürlich nicht. «Zum Glück» ist die benötigte Konfiguration berechenbar, ja sogar in einem intuitiven Sinne sehr regelmäßig, also jedenfalls endlich beschreibbar.

Banks (1971) gibt einen synchron arbeitenden deterministischen zweidimensionalen Zellularautomaten mit $M_1^{(2)}$ -Nachbarschaft an, der einen ebenfalls einen programmgesteuerten Universalrechner simuliert und dabei sogar mit nur *zwei* Zuständen auskommt. Die Konstruktionen sind dann komplizierter (siehe auch Toffoli und Margolus (1987, Abschnitt 5.5)), bedienen sich aber im wesentlichen der gleichen Prinzipien wie weiter vorne für WIREWORLD skizziert. Der Vorteil des Banks'schen Zellularautomaten ist, dass er die Erzeugung zusätzlicher «Drähte» erlaubt und man daher immer mit einer sogenannten endlichen Konfiguration¹ beginnen kann.

Codd (1968) gibt eine Lösung mit acht Zuständen für $H_1^{(2)}$ -Nachbarschaft an und zeigt, dass für diese Nachbarschaft jedenfalls gewisse Lösungen mit zwei Zuständen unmöglich sind.

Damit kommen wir zum zweiten Teil dieses Kapitels. Als allgemein akzeptiertes (und hoffentlich bekanntes) Modell für Berechenbarkeit nehmen wir Turingmaschinen her.

2.6 DEFINITION Eine *deterministische Turingmaschine* (kurz TM) mit einem Band und einem Kopf ist durch die folgenden Größen festgelegt:

1. die Zustandsmenge Q ,
2. das Bandalphabet B und
3. die Überföhrungsfunktion $\delta : Q \times B \rightarrow Q \times B \times \{-1, 0, 1\}$

Eine *Konfiguration* einer TM ist durch ein Tripel $c = (s, b, p) \in Q \times B^{\mathbb{Z}} \times \mathbb{Z}$ gegeben, das aktuellen Zustand, aktuelle Bandbeschriftung und aktuelle Kopfposition wiedergibt.

Ein *Schritt* gemäß der *Konfigurationsüberföhrungsfunktion* $\Delta : Q \times B^{\mathbb{Z}} \times \mathbb{Z} \rightarrow Q \times B^{\mathbb{Z}} \times \mathbb{Z}$ ist wie folgt festgelegt. Für $c = (s, b, p)$ und $\Delta(c) = (s', b', p')$ gilt:

1. $s' = \delta(s, b(p))[1]$,
2. für alle $i \in \mathbb{Z}$ ist $b'(i) = \begin{cases} b(i) & \text{falls } i \neq p \\ \delta(s, b(p))[2] & \text{falls } i = p \end{cases}$
3. $p' = p + \delta(s, b(p))[3]$. ◇

2.7 SATZ. Zu jeder TM $T = (Q_T, B_T, \delta_T)$ existiert ein (synchron arbeitender deterministischer) eindimensionaler ZA $C = (Q_C, \delta_C, H_1^{(1)})$, der T ohne Zeitverlust schrittweise simuliert.²

2.8 BEWEISSKIZZE. (Man vergleiche auch Vollmar (1979, Abschnitt 2.4).) Wir benutzen die Kette der EA des eindimensionalen ZA, um darin (unter anderem) die Beschriftungen der Felder des TM-Bandes zu speichern: nebeneinanderliegende Zellen speichern nebeneinander stehende Symbole.

¹für die genaue Definition siehe nächstes Kapitel.

²Wir präzisieren den Simulationsbegriff hier nicht weiter.

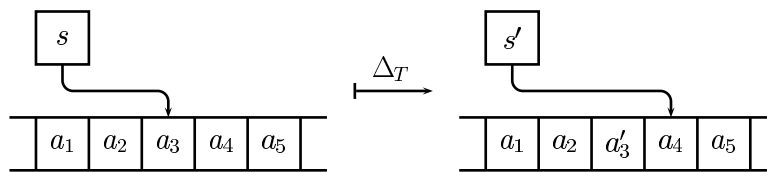
Die Zelle, die das gerade von der TM besuchte Feld repräsentiert, speichert außerdem den aktuellen Zustand der TM, während die anderen Zellen eine Markierung $_$ mit der Bedeutung «nicht besucht» tragen.

Wir setzen also $Q_C = (Q_T \cup \{_ \}) \times B_T$ und repräsentieren eine TM-Konfiguration $c_T = (s, b, p)$ durch die ZA-Konfiguration

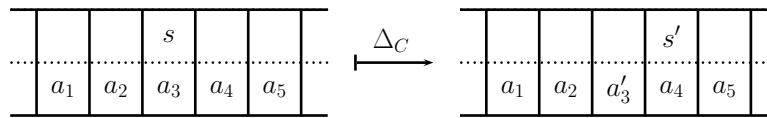
$$c_C : \mathbb{Z} \rightarrow Q_C$$

$$i \mapsto \begin{cases} (s, b(p)) & \text{falls } i = p \\ (_, b(i)) & \text{falls } i \neq p \end{cases}$$

Die lokale Überföhrungsfunktion δ_C des ZA muss nun so festgelegt werden, dass ein TM-Schritt



(falls etwa $\delta_T(s, a_3) = (s', a'_3, 1)$ ist) durch *einen einzigen* ZA-Schritt (der Einfachheit halber sind die $_$ alle nicht dargestellt)



simuliert werden kann.

Wir skizzieren für die wichtigen Fälle die Festlegung von δ_C ; die Argumente seien (von links nach rechts) der Zustand des linken Nachbarn, der eigene Zustand und der Zustand des rechten Nachbarn. Man setze für alle $a_l, a'_l, a_m, a'_m, a_r, a'_r \in B_T$ und alle $s, s' \in Q_T$:

$$\delta_C(_, a_l, (_, a_m), (_, a_r)) = (_, a_m)$$

$$\delta_C((s, a_l), (_, a_m), (_, a_r)) = \begin{cases} (s', a_m) & \text{falls } \delta_T(s, a_l) = (s', a'_l, 1) \\ (_, a_m) & \text{sonst} \end{cases}$$

$$\delta_C((_, a_l), (s, a_m), (_, a_r)) = \begin{cases} (s', a'_m) & \text{falls } \delta_T(s, a_m) = (s', a'_m, 0) \\ (_, a'_m) & \text{falls } \delta_T(s, a_m) = (s', a'_m, \pm 1) \end{cases}$$

$$\delta_C((_, a_l), (_, a_m), (s, a_r)) = \begin{cases} (s', a_m) & \text{falls } \delta_T(s, a_r) = (s', a'_r, -1) \\ (_, a_m) & \text{sonst} \end{cases}$$

Alle anderen Kombinationen von Argumenten können bei der Simulation nicht auftreten. Für sie kann der Funktionswert von δ_C willkürlich festgelegt werden.

Eigentlich müsste nun (etwa per Induktion) bewiesen werden, dass diese Konstruktion das Gewünschte leistet. Das unterlassen wir an dieser Stelle, da es nicht lehrreich ist. ■

2.9 ÜBUNG. Man überlege sich entsprechende Konstruktionen für den Fall von TM mit *zwei* Bändern und je einem Kopf darauf sowie anschließend für den Fall von TM mit *zwei* Köpfen auf nur einem Band, bei denen der ZA jeweils höchstens um einen konstanten Faktor langsamer ist als die TM.

Was ist mit der «Umkehrung» von Satz 2.7? Eine TM kann nicht in endlich vielen Schritten unendlich viele ZA-Aktivitäten simulieren. Braucht sie aber auch nicht: denn mit den Eingaben sind immer auch in allen folgenden Konfigurationen nur *endliche* Ausschnitte von «Interesse». Damit werden wir uns im nächsten Kapitel befassen.

Zusammenfassung

1. Es gibt Zellularautomaten, die bei jeweils geeignet gewählten Anfangskonfigurationen einen klassischen Universalrechner simulieren können.
2. Zellularautomaten können die gleichen Funktionen berechnen wie Turingmaschinen.

Literatur

- Banks, E. R. (Jan. 1971). *Information Processing and Transmission in Cellular Automata*. Techn. Ber. MIT/LCS/TR-81. PhD. MIT Laboratory for Computer Science (siehe S. 10).
- Codd, E. F. (1968). *Cellular Automata*. New York: Academic Press (siehe S. 10).
- Dewdney, A. K. (1990). «Draht- und Teppichwelten». In: *Spektrum der Wissenschaften* Sonderheft 10. Erschien ursprünglich in *Scientific American*, January 1990., S. 36–39 (siehe S. 7).
- 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. 10).
- Vollmar, Roland (1979). *Algorithmen in Zellularautomaten*. Stuttgart: Teubner (siehe S. 10).