

Ein einfaches universales Logikelement und Zellularautomaten für Rechnerreversibilität

Proseminar Zellularautomaten und diskrete komplexe Systeme

Autor: Dennis Felsing
Betreuer: Thomas Worsch

6. Juli 2011

Zusammenfassung

Diese schriftliche Ausarbeitung befasst sich mit einem einfachen Logikelement, dem „Drehelement“. Es wird die Universalität dieses Elements dadurch gezeigt, dass sich eine reversible Turingmaschine als Schaltung nur dieser Elemente konstruieren lässt. Anschließend wird vorgestellt wie sich dieses Element und Schaltungen daraus in einem einfachen reversiblen Zellularautomaten implementieren lassen.

Inhaltsverzeichnis

1	Einleitung	2
2	Voraussetzungen	2
2.1	Reversible Mealy-Automaten	2
2.2	Drehelemente	2
2.3	Drehelement-Schaltungen	4
2.4	Drehelement-Spalten	5
3	Reversible Turingmaschinen aus Drehelementen	7
3.1	Reversible Turingmaschinen	7
3.2	Konstruieren einer Bandeinheit	8
3.3	Konstruieren einer Steuereinheit	11
3.4	Konstruieren einer reversiblen Turingmaschine	14
4	Reversible Zellularautomaten für Drehelemente	14
5	Schluss	15

1 Einleitung

Diese Proseminararbeit stützt sich in großen Teilen auf [1].

Die Berechnungen heutiger Rechner funktionieren mit irreversibler Logik. Das heißt bei Operationen kann Information verloren gehen indem Bits gelöscht werden. Das Landauer-Prinzip besagt, dass für ein solches Löschen eines Bits physikalisch eine Wärmeabgabe von mindestens $kT \ln 2$ notwendig ist, wobei k die Boltzmann-Konstante ($\sim 10^{-23} \frac{J}{K}$) und T die absolute Temperatur in Kelvin ist. Um diese zwingende Wärmeentwicklung zu vermeiden hat Bennett die Rechnerreversibilität eingeführt. Dabei werden Berechnungen so ausgeführt, dass sie sich umkehren lassen. Es geht also keinerlei Information verloren und somit ist auch keine Wärmeabgabe notwendig.

Diese Arbeit stellt in Kapitel 2 ein reversibles Element vor, das sogenannte Drehelement, und zeigt wie sich aus diesem Schaltungen konstruieren lassen, die ebenfalls reversibel sind.

In Kapitel 3 wird die Turing-Vollständigkeit dieser Schaltungen bewiesen, indem ein Verfahren zur Konstruktion einer reversiblen Turingmaschine als Drehelement-Schaltung beispielhaft angewandt wird.

Kapitel 4 beschreibt einen einfachen und reversiblen Zellularautomaten und zeigt wie sich in diesen Drehelemente und deren Schaltungen implementieren lassen.

2 Voraussetzungen

2.1 Reversible Mealy-Automaten

Definition 1. Ein Reversibler Mealy-Automat (RMA) $M = (Q, \Sigma, \Gamma, q_1, \delta)$ ist festgelegt durch:

- eine endliche, nichtleere Zustandsmenge Q
- ein Eingabealphabet Σ
- ein Ausgabealphabet Γ
- einen Anfangszustand $q_1 \in Q$
- eine bijektive Zustandsübergangsfunktion $\delta : Q \times \Sigma \rightarrow Q \times \Gamma$

Eine bijektive Funktion ist eine Funktion, bei der jedes Element der Zielmenge genau einmal angenommen wird. Die Bijektivität der Übergangsfunktion bewirkt, dass man von jeder Kombination auszeitigem Zustand und letzter Ausgabe aus eindeutig den vorherigen Zustand und die letzte Eingabe bestimmen kann. In diesem Sinne bezeichnen wir einen solchen Automaten als „reversibel“

Eine Variation $M = (Q, \Sigma, \Gamma, \delta)$ ohne Anfangszustand wird der Einfachheit halber auch als RMA bezeichnet.

2.2 Drehelemente

Ein *Drehelement* (DE) ist ein logisches Element, dargestellt in Abbildung 1. Es verfügt über vier Eingänge $\{n, o, s, w\}$, vier Ausgänge $\{n', o', s', w'\}$ und zwei

Zustände namens H-Zustand und V-Zustand. Im H-Zustand befindet sich der Drehbalken im Zentrum der Darstellung des Drehelements in horizontaler, im V-Zustand in vertikaler Lage. Die Werte aller Ein- und Ausgänge sind entweder 0 oder 1, das heißt $(n, o, s, w), (n', o', s', w') \in \{0, 1\}^4$. Wir beschränken den Wertebereich der Eingabe durch $n + o + s + w \leq 1$ und der Ausgänge durch $n' + o' + s' + w' \leq 1$ jeweils auf $\{(0, 0, 0, 0), (1, 0, 0, 0), (0, 1, 0, 0), (0, 0, 1, 0), (0, 0, 0, 1)\}$. Wir fordern also, dass jeweils höchstens eine 1 als Eingang und Ausgang vorkommen kann. Das Verhalten des Drehelements für den Fall, dass an mehr als einem Eingang eine 1 anliegt, ist undefiniert.

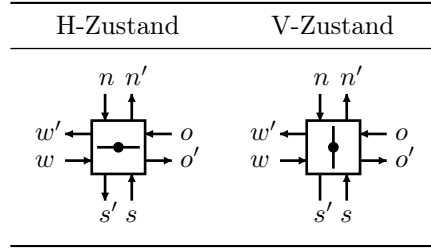
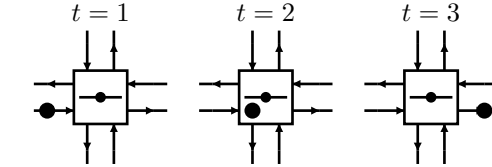


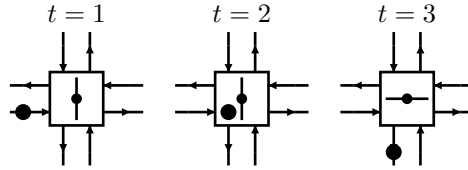
Abbildung 1: Zwei Zustände eines Drehelements

Die Funktionsweise eines Drehelements lässt sich folgendermaßen intuitiv interpretieren: Die Signale 1 und 0 stehen jeweils für die Existenz bzw. Nichtexistenz eines Teilchens. Im Zentrum des Drehelements befindet sich ein Drehbalken. Es können die folgenden Fälle auftreten:

Wenn kein Teilchen eingeht, passiert nichts mit dem Drehelement.



(a) Der parallele Fall (das heißt die Richtung, aus der das Teilchen kommt, ist parallel zum Drehbalken)



(b) Der orthogonale Fall

Abbildung 2: Operationen eines Drehelements

Wenn ein Teilchen von einem Eingang aus einer Richtung parallel zum Drehbalken kommt, so bewegt es sich geradeaus durch das Drehelement und verlässt dieses aus dem Ausgang der gegenüberliegenden Seite. Die Richtung des Drehbalken wird dadurch nicht beeinflusst. (Abbildung 2(a))

Wenn ein Teilchen von einem Eingang aus einer Richtung orthogonal zum

Drehbalken kommt, so trifft es auf den Drehbalken auf, dreht diesen um 90 Grad im Gegenuhrzeigersinn, biegt nach rechts ab und verlässt das Drehelement aus dem Ausgang zur Rechten. (Abbildung 2(b))

Formal lässt sich ein Drehelement als Reversibler Mealy-Automat M_{DE} definieren. Da die Eingabe $(0, 0, 0, 0)$ keinen Einfluss auf das Drehelement hat, lassen wir sie beim Eingabealphabet der Einfachheit halber weg. Desweiteren bezeichnen wir die Eingabe- und Ausgabealphabete als $\{n, o, s, w\}$ und $\{n', o', s', w'\}$ statt $\{(1, 0, 0, 0), (0, 1, 0, 0), (0, 0, 1, 0), (0, 0, 0, 1)\}$.

M_{DE} ist definiert durch

$$M_{DE} = (\{\boxed{\rightarrow}, \boxed{\uparrow}\}, \{n, o, s, w\}, \{n', o', s', w'\}, \delta_{DE})$$

Tabelle 1: Die Zustandsübergangsfunktion δ_{DE} des Drehelements M_{DE}

Aktueller Zustand	Eingabe			
	n	o	s	w
$\boxed{\rightarrow}$	$\boxed{\uparrow} w'$	$\boxed{\rightarrow} w'$	$\boxed{\uparrow} o'$	$\boxed{\rightarrow} o'$
$\boxed{\uparrow}$	$\boxed{\uparrow} s'$	$\boxed{\rightarrow} n'$	$\boxed{\uparrow} n'$	$\boxed{\rightarrow} s'$

mit der Zustandsübergangsfunktion δ_{DE} wie in Tabelle 1 dargestellt. Das Symbol $\boxed{\rightarrow}$ steht für den H-Zustand des Drehelements, in dem der Drehbalken horizontal ausgerichtet ist, $\boxed{\uparrow}$ für den V-Zustand. Wenn beispielsweise im Zustand $\boxed{\rightarrow}$ ein Teilchen von Eingang n kommt, dann ändert sich entsprechend Tabelle 1 der Zustand zu $\boxed{\uparrow}$ und das Teilchen verlässt das Drehelement durch w' .

Die Übergangsfunktion δ_{DE} ist bijektiv, da jede Kombination von neuem Zustand und Ausgabe durch genau eine Kombination von aktuellem Zustand und Eingabe erreicht werden kann. Somit ist ein Drehelement reversibel. Eine weitere interessante Eigenschaft des Drehelements ist die Bit-Erhaltung, das heißt die Anzahl der Einsen bleibt zwischen Eingängen und Ausgängen erhalten. Intuitiv interpretiert besagt dies, dass ein Teilchen weder ausgelöscht noch neu erschaffen wird.

2.3 Drehelement-Schaltungen

Eine *Drehelement-Schaltung* ist eine Schaltung nur aus Drehelementen, für die gilt: Jeder Ausgang eines Drehelements ist mit höchstens einem Eingang eines Drehelements (auch desselben) verbunden. Somit ist ein Ansteuern mehrerer Eingänge von einem Ausgang aus, ein sogenanntes Fan-Out, nicht möglich. Dafür überträgt sich aber die Reversibilität der Elemente auf die Schaltung und somit lässt sich auch diese als Reversibler Mealy-Automat formulieren.

Ein Beispiel für eine Drehelement-Schaltung ist in Abbildung 3 dargestellt. Als RMA schreibt sie sich als

$$M_{Bsp} = (\{\boxed{\rightarrow\rightarrow}, \boxed{\rightarrow\uparrow}, \boxed{\uparrow\rightarrow}, \boxed{\uparrow\uparrow}\}, \{a\}, \{b, c\}, \delta_{Bsp}).$$

Die Zustandsübergangsfunktion δ_{Bsp} ist in Tabelle 2 dargestellt. Analog zum Drehelement bedeutet das Eingangssymbol a , dass am Eingang a eine 1 anliegt, und die Ausgangssymbole b und c jeweils dass an Ausgang b bzw. c eine 1 anliegt.

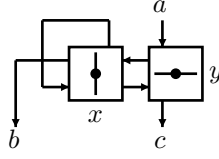


Abbildung 3: Beispiel einer Drehelement-Schaltung

Tabelle 2: Die Zustandsübergangsfunktion δ_{Bsp} der Drehelement-Schaltung M_{Bsp}

Eingabe	
Aktueller Zustand	a
	c
	b
	c
	c

Das erste Zeichen eines Zustands beschreibt den Zustand des Drehelements x , das zweite Zeichen den von y . Der Zustand bedeutet also beispielsweise, dass x im Zustand (V-Zustand) und y im Zustand (H-Zustand) ist. Wenn in diesem Zustand ein Teilchen vom Eingang a kommt, dann ändert sich, der Zustand von x zu , y bleibt im Zustand und das Teilchen verlässt die Drehelement-Schaltung durch c . Man kann sich dies überlegen, indem man das Verhalten der einzelnen Drehelemente betrachtet: Das Teilchen trifft von a aus orthogonal zum Drehbalken in y auf und ändert somit den Zustand zu und bewegt sich nach links weiter. Es trifft dann analog auf x auf, ändert den Zustand zu und verlässt das x nach oben. Anschließend wird das Teilchen von links parallel zum Drehbalken in x auf, bewegt sich also ohne Änderung des Zustands weiter. Beim letzten Auftreffen auf y wird der Zustand zu geändert und das Teilchen verlässt die Schaltung über Ausgang c .

2.4 Drehelement-Spalten

Wir betrachten eine besondere Drehelement-Schaltung namens Drehelement-Spalte, um später zu erklären, wie sich daraus ein Band einer Turingmaschine aufbauen lässt.

Eine Drehelement-Spalte liegt in Abbildung 4(a) vor. Sie besteht aus $k + 1$ Drehelementen, $2k$ Eingänge, und $2k$ Ausgänge. Alle Drehelemente außer dem untersten sind anfangs im V-Zustand. Das unterste Drehelement, mit x markiert, kann sich entweder im V-Zustand oder H-Zustand befinden. Ein Signal darf immer nur an höchstens einem Eingang eingehen. Damit können wir eine Drehspalte als Reversiblen Mealy-Automaten M_{DES} betrachten und formulieren als

$$M_{DES} = (\{\langle \text{H-Zustand}, \text{V-Zustand} \rangle\}, \{l_1, l_2, \dots, l_k, r_1, r_2, \dots, r_k\}, \{l'_1, l'_2, \dots, l'_k, r'_1, r'_2, \dots, r'_k\}, \delta_{DES})$$

mit der Zustandsüberföhrungsfunktion δ_{DES} , wie sie in Abbildung 4(b) aufgeführt ist. Die möglichen Zustände $\{\langle \text{H-Zustand}, \text{V-Zustand} \rangle\}$ stehen dabei für den Zustand des

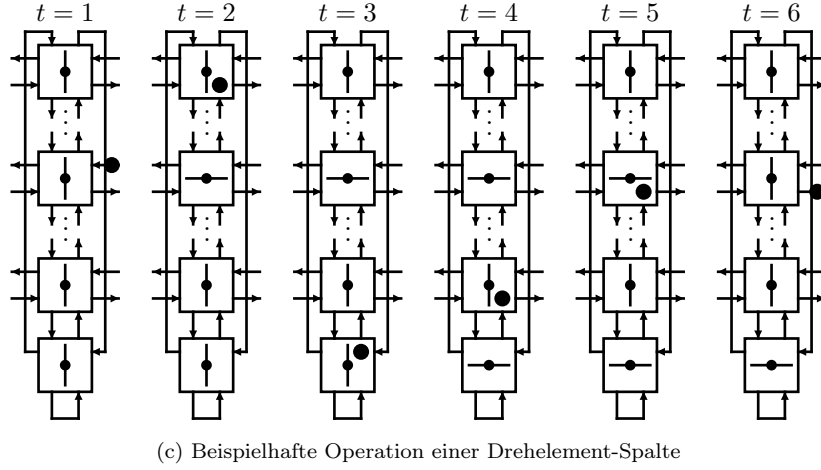
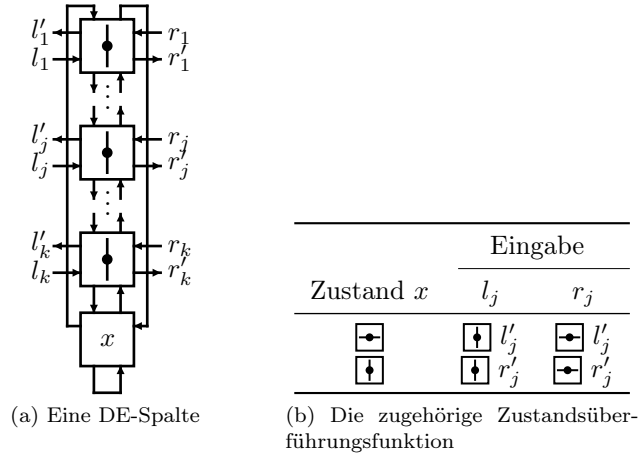


Abbildung 4: DE-Spalte

untersten, mit x bezeichneten, Drehelements. Die anderen Drehelemente können ihren Zustand aus folgendem Grund nicht ändern: Ein eingehendes Signal trifft immer zuerst auf das Drehelement, mit dem der Eingang direkt verbunden ist und setzt dieses vom Zustand $\begin{smallmatrix} \blacksquare \\ \downarrow \end{smallmatrix}$ in den Zustand $\begin{smallmatrix} \blacksquare \\ \leftarrow \end{smallmatrix}$. Durch die anderen Drehelemente, außer x , kann das Signal sich nur geradeaus durchbewegen, da es nur über Eingänge, die parallel zu dessen Drehbalken liegen, ankommen kann. Das Signal kann also die Schaltung nur verlassen indem es das Drehelement, dessen Zustand er anfangs geändert hat, erneut ändert und am Ausgang links oder rechts davon austritt. Auch dieses Drehelement ist somit nach Verlassen des Signals wieder im Ausgangszustand.

Die Korrektheit von δ_{DES} lässt sich wie folgt überprüfen: Wir nehmen zum Beispiel als Anfangszustand von x $\begin{smallmatrix} \blacksquare \\ \downarrow \end{smallmatrix}$ und als Eingang ein Signal auf r_j . In Abbildung 4c sieht man, wie sich das Signal dann in der Schaltung fortbewegt.

Im Folgenden schreiben wir $x = 0$, wenn x sich im Zustand $\begin{smallmatrix} \blacksquare \\ \downarrow \end{smallmatrix}$ befindet, und $x = 1$, wenn x sich im Zustand $\begin{smallmatrix} \blacksquare \\ \leftarrow \end{smallmatrix}$ befindet. Man kann ein Drehelement als Speicherzelle für ein Bit ansehen. Das Drehelement x speichert dieses durch

seinen Zustand. Mit einem Signal von l_j lässt sich $x = 0$ setzen. Wenn vorher schon $x = 0$ war, dann wird ein Signal auf r'_j ausgegeben, wenn jedoch vorher $x = 1$ war, dann auf l'_j .

3 Reversible Turingmaschinen aus Drehelementen

3.1 Reversible Turingmaschinen

Wir definieren zuerst eine Einband-Turingmaschine. Dazu wird der Quadrupel-Formalismus [2] einer Turingmaschine verwendet. Dieser ermöglicht es einfach die Reversibilität einer Turingmaschine zu definieren und ist ebenso mächtig wie die gebräuchlichere Quintupel-Schreibweise. Eine Umwandlung einer Quintupel- in eine Quadrupel-Turingmaschine ist in Abbildung 6 aufgezeigt.

Definition 2. Eine Einband-Turingmaschine (TM) $T = (Q, S, q_0, q_f, s_0, \delta)$ ist festgelegt durch:

- eine endliche, nichtleere Zustandsmenge Q
- ein Bandalphabet S
- einen Anfangszustand $q_0 \in Q$
- einen Endzustand $q_f \in Q$
- ein Blanksymbol $s_0 \in S$
- eine partielle Zustandsübergangsfunktion
 $\delta \subseteq (Q \times S \times S \times Q) \cup (Q \times \{/\} \times \{-, 0, +\} \times Q)$

Die Elemente von δ werden als Quadrupel bezeichnet und sind entweder von der Form $[q_r, s, s', q_t] \in (Q \times S \times S \times Q)$ oder $[q_r, /, d, q_t] \in (Q \times \{/\} \times \{-, 0, +\} \times Q)$. Dabei stehen die Symbole $-, 0, +$ jeweils für eine Kopfbewegung nach links, keine Kopfbewegung, bzw. eine Kopfbewegung nach rechts. Der Quadrupel-Formalismus trennt somit Zustandsübergänge, bei denen sich die Bandbeschriftung an der Kopfposition ändert und solche bei denen sich die Kopfposition ändert. Ein Quadrupel $[q_r, s, s', q_t]$ bedeutet: Falls T sich in Zustand q_r befindet und an der Kopfposition das Symbol s liest, wird s' an die Kopfposition geschrieben und in den Zustand q_t gewechselt. Ein Quadrupel $[q_r, /, d, q_t]$ bedeutet: Falls T sich in Zustand q_r befindet, wird die Kopfposition in Richtung d bewegt und in den Zustand q_t gewechselt.

Seien $\alpha_1 = [p_1, b_1, c_1, p'_1]$ und $\alpha_2 = [p_2, b_2, c_2, p'_2]$ Quadrupel, also $\alpha_1, \alpha_2 \in \delta$. α_1 und α_2 überlappen im Definitionsbereich genau dann, wenn

$$p_1 = p_2 \wedge [b_1 = b_2 \vee b_1 = / \vee b_2 = /].$$

Damit α_1 und α_2 sich nicht im Definitionsbereich überlappen, muss sich entweder der Anfangszustand unterscheiden oder α_1 ein anderes Symbol lesen als α_2 und keines der beiden Quadrupel eine Kopfbewegung beschreiben. Somit gibt es ohne Überlappung im Definitionsbereich keine Situation in der sich sowohl α_1 als auch α_2 anwenden lassen.

α_1 und α_2 überlappen im Wertebereich genau dann, wenn

$$p'_1 = p'_2 \wedge [c_1 = c_2 \vee b_1 = \neg \vee b_2 = \neg].$$

Die Überlappung im Wertebereich ist analog zur Überlappung im Definitionsbereich, hängt aber vom End- statt Anfangszustand und geschriebenem statt gelesenen Symbol ab. Das heißt, damit α_1 und α_2 sich nicht im Wertebereich überlappen, muss sich entweder der Endzustand unterscheiden oder α_1 ein anderes Symbol schreiben als α_2 und keines der beiden Quadrupel eine Kopfbewegung beschreiben. Ohne Überlappung im Wertebereich kann zwischen α_1 und α_2 entschieden werden, welches Quadrupel angewendet wurde um T in die aktuelle Konfiguration zu bringen.

Ein Quadrupel α ist deterministisch (in δ) genau dann, wenn es kein anderes Quadrupel in δ gibt mit dem α sich im Definitionsbereich überlappt. Ein Quadrupel α ist reversibel (in δ) genau dann, wenn es kein anderes Quadrupel in δ gibt mit dem α sich im Wertebereich überlappt.

T ist deterministisch genau dann, wenn jedes Quadrupel in δ deterministisch ist. T ist reversibel genau dann, wenn jedes Quadrupel in δ reversibel ist. Im Folgenden werden nur deterministische Turingmaschinen betrachtet.

Theorem 1. Für jede Einband-Turingmaschine gibt es eine reversible (semiunendliche) Einband-Zwei-Symbol-Turingmaschine, welche die Erstere simuliert. [3]

Der Beweis des Theorems enthält eine Konstruktionsanleitung für eine solche reversible Einband-Zwei-Symbol-Turingmaschine. Somit lässt sich jede Turingmaschine in eine solche umsetzen. Es genügt also zu zeigen, dass sich eine reversible Einband-Zwei-Symbol-Turingmaschine als Drehelement-Schaltung konstruieren lässt um die Turing-Vollständigkeit von Drehelement-Schaltungen zu beweisen.

Im Folgenden werden eine einzelne Bandzelle des semiunendlichen Bandes und die Steuereinheit einer reversiblen Turingmaschine konstruiert.

3.2 Konstruieren einer Bandeinheit

Ein *Bandzellenmodul* (BZ-Modul) simuliert eine einzelne Bandzelle einer reversiblen Einband-Zwei-Symbol-Turingmaschine, wie sie soeben eingeführt wurde. Das BZ-Modul kann ein Symbol 0 oder 1, das auf die Zelle geschrieben ist, und eine Information ob der Bandkopf sich auf der Bandzelle befindet oder nicht, speichern.

Wir bezeichnen Signale, die für Befehle von der Steuereinheit an die Bandzelle über der sich der Kopf befindet stehen, als $\mathbb{B} := \{R, W, SL, SR, E0, E1\}$, ihre Bedeutungen sind in Tabelle 3 aufgelistet. Signale, die zur Kommunikation von Bandzellen untereinander dienen, bezeichnen wir als $\mathbb{I} := \{SRI, SLI\}$, Tabelle 4 enthält ihre Bedeutungen. Schließlich bezeichnen wir Signale, welche zur Benachrichtigung über Abschluss eines Befehls an die Steuereinheit berichten, als $\mathbb{C} := \{Rc0, Rc1, Wc, SLc, SRe, Ec\}$, Tabelle 5 enthält ihre Bedeutungen. Es gibt keine weiteren Signale.

Ein BZ-Modul lässt sich als Reversibler Mealy-Automat M_{BZ} wie folgt de-

Tabelle 3: Signale, die als Befehle der Steuereinheit an Bandzelle an Kopfposition dienen

Signal	Bedeutung
R	Lies das Symbol.
W	Schreibe ein komplementäres Symbol. (Falls das derzeitige Symbol 1 ist, schreibe 0, ansonsten schreibe 1.)
SL	Bewege den Kopf nach links.
SR	Bewege den Kopf nach rechts.
$E0$	Lösche die Information 0 der Steuereinheit reversibel durch Weitergeben des Symbols 0 an die Kopfposition, die 0 als Zustand gespeichert haben muss.
$E1$	Lösche die Information 1 der Steuereinheit reversibel durch Weitergeben des Symbols 1 an die Kopfposition, die 1 als Zustand gespeichert haben muss.

Tabelle 4: Signale, die zur Kommunikation benachbarter Bandzellen dienen

Signal	Bedeutung
SLI	Bewege den Kopf von linkem Nachbarn auf diese Zelle.
SRI	Bewege den Kopf von rechtem Nachbarn auf diese Zelle.

finieren:

$$M_{BZ} = (Q_{BZ}, \Sigma_{BZ}, \Gamma_{BZ}, \delta_{BZ})$$

$$Q_{BZ} = \{(h, s) \mid h, s \in \{0, 1\}\}$$

$$\Sigma_{BZ} = \mathbb{B} \cup \mathbb{I} \cup \mathbb{C}$$

$$\Gamma_{BZ} = \{x' \mid x \in \Sigma_{BZ}\}$$

δ_{BZ} ist wie folgt definiert, mit $s \in \{0, 1\}$ und $y \in \mathbb{B} \cup \mathbb{C}$:

$$\delta_{BZ}((0, s), y) = (0, s, y') \quad (1)$$

$$\delta_{BZ}((0, s), SRI) = (1, s, SRc') \quad (2)$$

$$\delta_{BZ}((0, s), SLI) = (1, s, SLc') \quad (3)$$

$$\delta_{BZ}((1, 0), R) = (1, 0, Rc0') \quad (4)$$

$$\delta_{BZ}((1, 1), R) = (1, 1, Rc1') \quad (5)$$

$$\delta_{BZ}((1, 0), W) = (1, 1, Wc') \quad (6)$$

$$\delta_{BZ}((1, 1), W) = (1, 0, Wc') \quad (7)$$

$$\delta_{BZ}((1, s), SR) = (0, s, SRI') \quad (8)$$

$$\delta_{BZ}((1, s), SL) = (0, s, SLI') \quad (9)$$

$$\delta_{BZ}((1, 0), E0) = (1, 0, Ec') \quad (10)$$

$$\delta_{BZ}((1, 1), E1) = (1, 1, Ec') \quad (11)$$

Tabelle 5: Signale, die als Antwort von Bandzelle an Steuereinheit dienen

Signal	Bedeutung
<i>Rc0</i>	Eine Leseoperation ist fertig mit Ergebnis 0.
<i>Rc1</i>	Eine Leseoperation ist fertig mit Ergebnis 1.
<i>Wc</i>	Eine Schreiboperation ist fertig.
<i>SLc</i>	Bewegen des Kopfs nach links ist fertig.
<i>SRc</i>	Bewegen des Kopfs nach rechts ist fertig.
<i>Ec</i>	Eine reversible Löschoption ist fertig.

M_{BZ} hat die Zustandsmenge $\{(h, s) \mid h, s \in \{0, 1\}\}$. Der Zustand (h, s) repräsentiert, dass das Symbol s auf der Bandzelle steht und dass der Bandkopf auf der Zelle ist, falls $h = 1$, oder nicht. Zu jeder Eingabe $x \in \Sigma_{BZ}$ gibt es eine korrespondierende Ausgabe $x' \in \Gamma_{BZ}$.

Die Fälle von δ_{BZ} lassen sich wie folgt unterscheiden:

1. $h = 0$:
 - (a) Signal an Eingang $y \in \mathbb{B} \cup \mathbb{C}$: Das Signal verlässt das BZM durch y' ohne seinen Zustand zu ändern. Das Signal war entweder ein Befehl an das BZM unter dem Kopf oder eine Antwort an die Steuereinheit, also auf jeden Fall nicht an dieses BZM gerichtet. Deshalb wird es einfach weitergeleitet. (1)
 - (b) Signal an Eingang *SRI* (bzw. *SLI*): Die Kopfposition h wird auf 1 gesetzt und verlässt als Abschlussignal für die Kopfverschiebung nach rechts (links) das BZM durch *SRc'* (*SLc'*). Der Kopf wurde vom linken (rechten) Nachbarn auf diese Zelle verschoben. (2), (3)
2. $h = 1$:
 - (a) Signal an Eingang *R* und Zustand $s = 0$ (bzw. $s = 1$): Das Signal verlässt das BZM durch *Rc0'* (*Rc1'*). Dadurch wird der aktuelle Zustand ausgelesen. (4), (5)
 - (b) Signal an Eingang *W* und $s = 0$ (bzw. $s = 1$): Es wird s auf 1 (0) gesetzt und das Signal verlässt das BZM durch *Wc'*. Dadurch wird eine komplementäre Schreiboperation ausgeführt. (6), (7)
 - (c) Signal an Eingang *SR* (bzw. *SL*): Die Kopfposition h wird auf 0 gesetzt und das Signal verlässt das BZM durch *SRI'* (*SLI'*). (8), (9)
 - (d) Signal an Eingang *E0* (bzw. *E1*) und $s = 0$ ($s = 1$): Das Signal verlässt das BZM durch *Ec'*. Dadurch wird ein „reversibler Löschvorgang“ eines Informationsbits ausgeführt indem das Symbol an der Kopfposition referenziert wird und die selbe Rückgabe erfolgt, unabhängig davon ob 0 oder 1 referenziert wurde. (10), (11)

Alle anderen Möglichkeiten sind undefiniert, zum Beispiel $\delta_{BZ}((1, 0), E1)$. Dadurch würde versucht die Information 1 zu löschen, auf der Zelle steht jedoch eine 0.

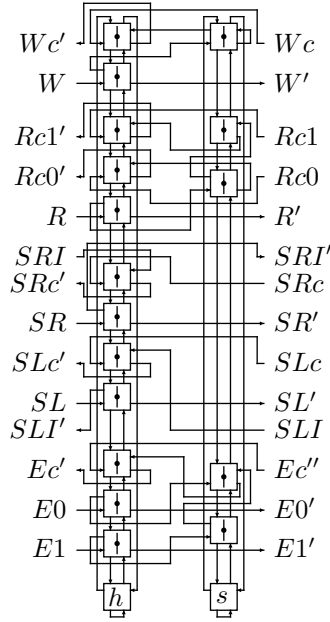


Abbildung 5: Ein Bandzellenmodul: Realisierung einer Bandzelle M_{BZ} als eine Drehelement-Schaltung.

Abbildung 5 zeigt ein Bandzellenmodul, eine Realisierung einer Bandzelle M_{BZ} als eine Drehelement-Schaltung. Sie ist aufgebaut aus zwei Drehelement-Spalten, die so miteinander verbunden sind, dass sie genau δ_{BZ} implementieren.

Ein Bandzellenmodul besteht aus zwei Drehelement-Spalten. Die linke Spalte speichert in h ob der Kopf auf der Zelle ruht ($h = 1$) oder nicht ($h = 0$). Die rechte Spalte speichert in s das Symbol auf der Zelle. Die Übereinstimmung des Bandzellenmoduls mit M_{BZ} lässt sich verifizieren indem alle gültigen Kombinationen von Eingängen und Zuständen durchprobiert werden.

Den Schaltkreis einer ganzen Bandeinheit erhält man durch unendlich viele Bandzellen, die man in einer Reihe anordnet und deren Eingänge und Ausgänge man wie im rechten Teil von Abbildung 9 zwischen benachbarten Bandzellen verbindet. Die Ansteuerung der Bandeinheit erfolgt dann über die Engänge R , W , SL , SR , $E0$, und $E1$ des Bandzellenmoduls ganz links.

3.3 Konstruieren einer Steuereinheit

Die Konstruktion eines Steuereinheitsmoduls führen wir beispielhaft an einer Turingmaschine T_{2n} aus, die für eine unär kodierte Zahl auf dem Band die Funktion $f(n) = 2n$ berechnet, wie zum Beispiel in Abbildung 7. Dabei muss die Kopfposition am Anfang auf einer 0 vor der eigentlichen Eingabe n stehen. Das Ergebnis wird hinter die Eingabe, getrennt von einer 0, geschrieben. Nach Abschluss der Berechnung befindet sich der Kopf auf der trennenden 0. In Abbildung 6(a) ist eine solche Turingmaschine in gewöhnlicher Notation (Lesen, Schreiben, und Kopfbewegung in einem Übergang möglich) spezifiziert. In Abbildung 6(b) ist dieselbe Turingmaschine in der in Definition 2 eingeführten Weise dargestellt.

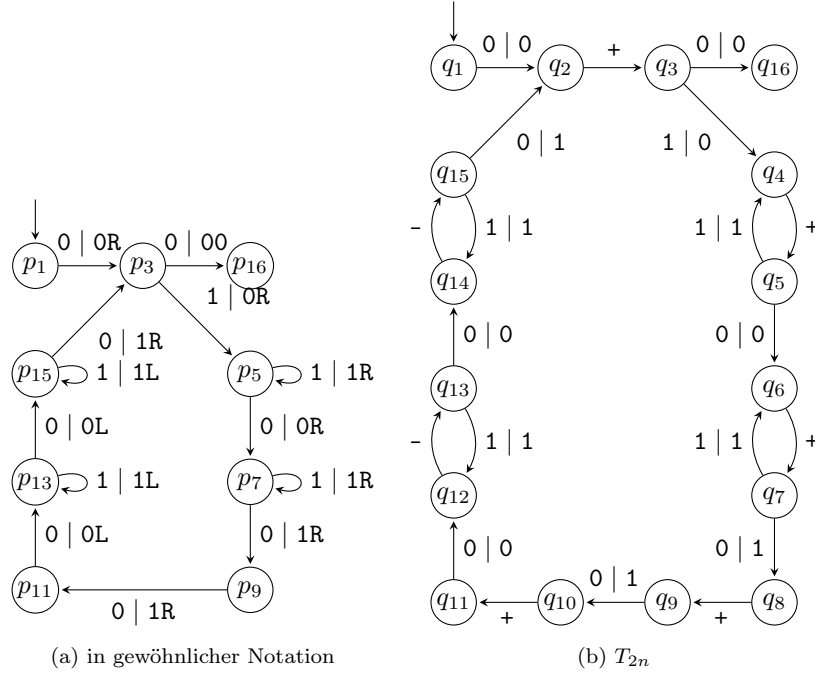


Abbildung 6: Turingmaschine zur Berechnung von $f(n) = 2n$

$$T_{2n} = (\{q_1, \dots, q_{16}\}, \{0, 1\}, q_1, q_{16}, 0, \delta_{2n})$$

mit den folgenden Quadrupeln als δ_{2n} :

$[q_1, 0, 0, q_2]$	$[q_4, /, +, q_5]$	$[q_8, /, +, q_9]$	$[q_{12}, /, -, q_{13}]$
$[q_2, /, +, q_3]$	$[q_5, 0, 0, q_6]$	$[q_9, 0, 1, q_{10}]$	$[q_{13}, 0, 0, q_{13}]$
$[q_3, 0, 0, q_{16}]$	$[q_5, 1, 1, q_4]$	$[q_{10}, /, +, q_{11}]$	$[q_{13}, 1, 1, q_{12}]$
$[q_3, 1, 0, q_4]$	$[q_6, /, +, q_7]$	$[q_{11}, 0, 0, q_{12}]$	$[q_{14}, /, -, q_{15}]$
	$[q_7, 0, 1, q_8]$		$[q_{15}, 0, 1, q_2]$
	$[q_7, 1, 1, q_6]$		$[q_{15}, 1, 1, q_{14}]$

Es lässt sich nachprüfen, dass T_{2n} reversibel ist, indem man nachprüft, dass keine zwei verschiedenen Quadrupel in δ_{2n} sich im Wertebereich überlappen.

Ein Steuereinheitsmodul für T_{2n} ist in Abbildung 8 dargestellt. Die Drehelemente sind in einer Matrix angeordnet, wobei jede Spalte mit einem Zustand von T_{2n} und jede Zeile mit einer der fünf Operationen, und zwar Schreiben, Lesen, Kopfbewegung nach rechts, Kopfbewegung nach links, reversibles Löschen, korrespondiert. Alle Drehelemente sind anfangs im H-Zustand.

Betrachten wir die möglichen Operationen, so fällt auf, dass es Inverse gibt. Eine Operation ist zu einer anderen invers, wenn sie deren Aktion rückgängig macht.

Das Inverse der Kopfbewegung nach links ist die Kopfbewegung nach rechts. Die Schreiboperation ist ihr eigenes Inverses. Das Inverse der Leseoperation ist

q_1	0	1	1	1	0	0	0	0	0	0	0	0
\vdots												
q_{16}	0	1	1	1	0	1	1	1	1	1	1	0

Abbildung 7: Berechnung der Funktion $f(n) = 2n$ von einer reversiblen Turingmaschine T_{2n} . Eine vollständige Auflistung aller Schritte befindet sich im Anhang.

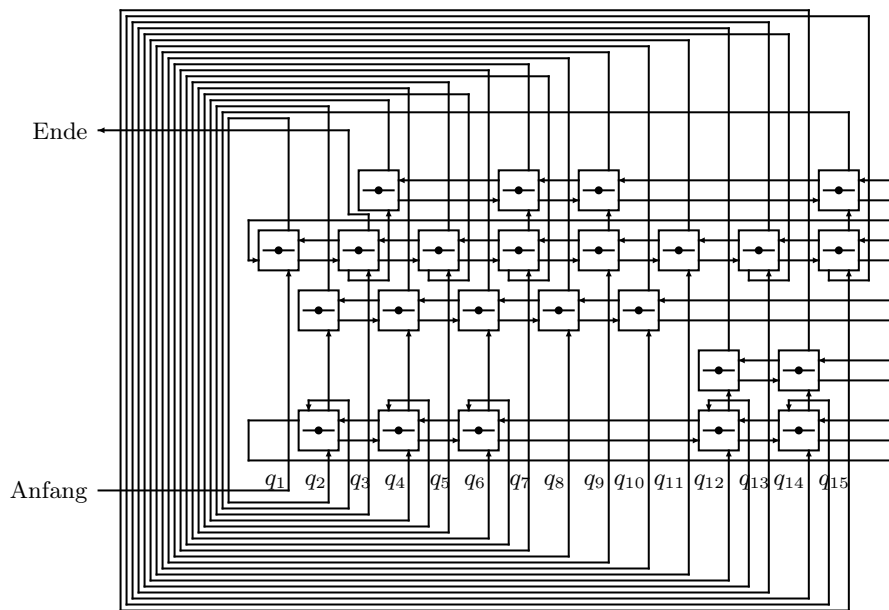


Abbildung 8: Das Steuereinheitsmodul für T_{2n}

die Löschoperation:

Wenn ein Zeichen gelesen wird, dann verlässt das Signal abhängig vom gelesenen Zeichen die Bandzelle aus einem anderen Ausgang. Um dies rückgängig zu machen, müssen die zwei Leitungen zu einer zusammengeführt werden. Ein einfaches Zusammenführen zweier Leitungen ist jedoch in Dreielement-Schaltungen nicht erlaubt, da ansonsten die Reversibilität nicht erhalten wäre. Wir benötigen jedoch unbedingt eine solche Operation, denn ansonsten würden mehrere Übergänge in einen Zustand des Zustandsautomaten (z.B. q_1 und q_{15} nach q_2 in Abbildung (b)) erfordern, dass man den Folgezustand dupliziert. Dadurch würde die Steuereinheit unendlich groß werden.

Um somit zu „vergessen“ ob das Zeichen 0 oder 1 gelesen wurde, wird dieses an die Bandzelle an Kopfposition übergeben und „reversibel gelöscht“.

3.4 Konstruieren einer reversiblen Turingmaschine

Durch richtiges Verbinden des Steuereinheitsmoduls mit dem Bandmodul erhält man eine Drehelement-Schaltung für eine gegebene reversible Turingmaschine. Abbildung 9 zeigt die Drehelement-Schaltung für T_{2n} . Durch ein Signal bei „Anfang“ beginnt sie zu berechnen. Nach der Berechnung verlässt das Signal die Schaltung durch die mit „Ende“ beschriftete Leitung.

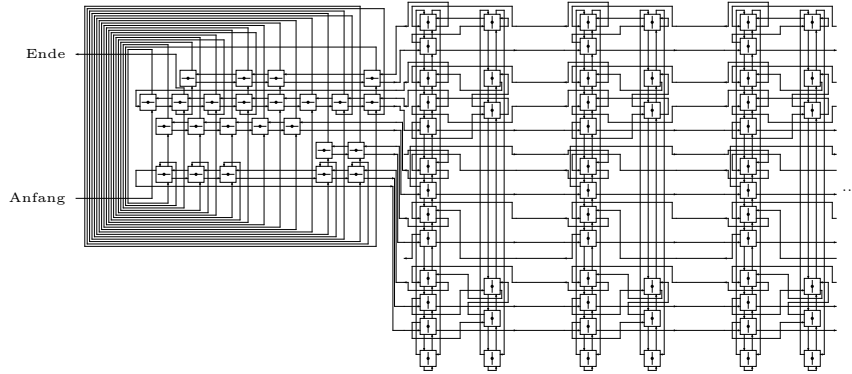


Abbildung 9: Die vollständige Drehelement-Schaltung T_{2n}

4 Reversible Zellularautomaten für Drehelemente

In [4] wurde ein reversibler partitionierter Zellularautomat P_3 vorgeschlagen. Bei diesem unterteilt sich jede Zelle in vier Teile, wie in Abbildung 10, die jeweils 3 Zustände, dargestellt als \square , \bullet , und \square , annehmen können. Der Zustand einer Zelle ist dabei genau von den anliegenden Teilen seiner benachbarten Zellen abhängig, siehe Abbildung 11. P_3 verfügt über die in Abbildung 12 dargestellten Regelschemata. Die Regeln des P_3 ergeben sich aus diesen durch Rotation um $\frac{\pi}{2}$, π , und $\frac{3\pi}{2}$. Regel (m) wird für die nicht durch (a)-(l) und deren Rotationen spezifizierten Übergänge verwendet. P_3 ist reversibel, da es keine zwei verschiedenen Regeln gibt, welche ein Feld in den selben Zustand bringen.

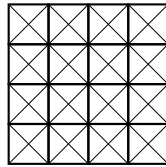


Abbildung 10: P_3

Ein Drehelement lässt sich in P_3 entsprechend Abbildung 13 integrieren. Signale werden als einzelnes \bullet in einem Feld notiert, konkret also $\begin{smallmatrix} \bullet & \diagdown \\ \diagup & \end{smallmatrix}$, $\begin{smallmatrix} \bullet & \diagup \\ \diagdown & \end{smallmatrix}$, $\begin{smallmatrix} \bullet & \diagdown \\ \diagdown & \end{smallmatrix}$, und $\begin{smallmatrix} \bullet & \diagup \\ \diagup & \end{smallmatrix}$. Zu beachten ist dabei, dass ein Signal nur zu jedem zweiten Zeitpunkt das Drehelement betreten darf, ansonsten verhält sich dieses im orthogonalen

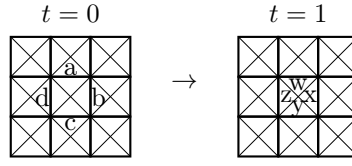


Abbildung 11: Zustandsübergänge im P_3

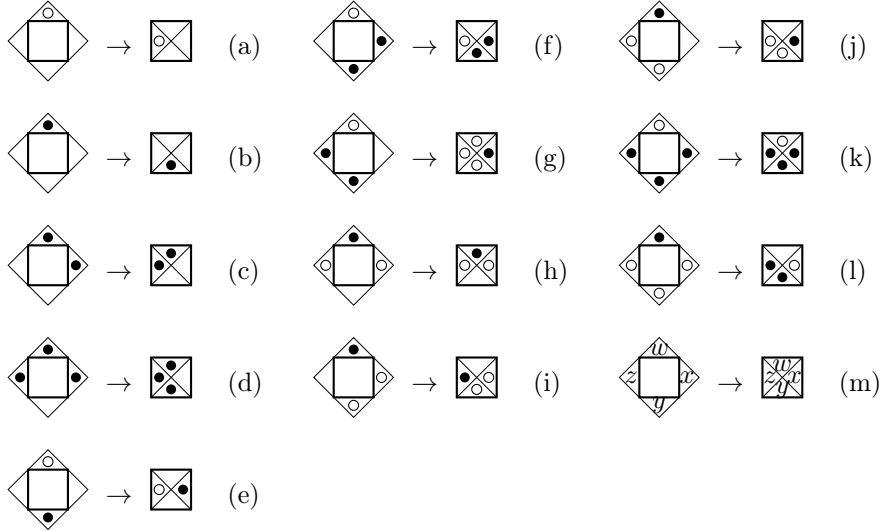


Abbildung 12: Die 13 Regelschemata des P_3

Fall falsch. Dadurch müssen auch die Wege zwischen Drehelementen in einer Drehelement-Schaltung in P_3 immer gerader Länge sein. In P_3 lassen sich Signale nicht kreuzen, da bei einer Kollision die Richtung eines Signals verändert wird. (Regeln (c) und (d))

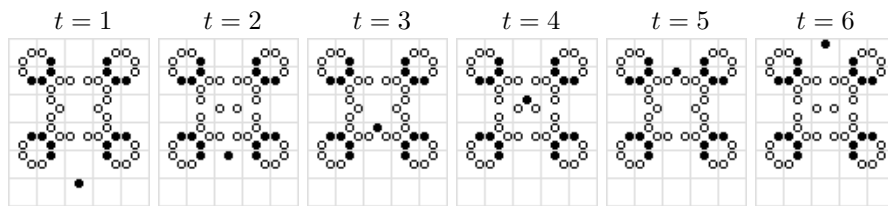
Mithilfe weiterer Elemente zur Lenkung von Signalen lassen sich auch Drehelement-Schaltung in P_3 realisieren. Im Gegensatz zum Drehelement sind diese verändern sich diese nicht und können auch zu jedem Zeitpunkt angesteuert werden. In Abbildung 14 befindet sich oben im Zentrum ein Rechtsdreher, der eingehende Signale immer nach rechts dreht. Unten im Zentrum ist ein Reflektor, der Signale in ihre Ursprungsrichtung zurückschickt. Die vier Elemente in den Ecken sind Links-Rechts-Dreher, die ein auf die linke Seite treffendes Signal, nach links umlenken und ein auf die rechte Seite treffendes Signal nach rechts. Auf der rechten Seite in der Mitte befindet sich das Eingangssignal.

Somit sind auch reversible Turingmaschine, wie in Abschnitt soundso als Drehelement-Schaltungen konstruiert, in P_3 abbildbar.

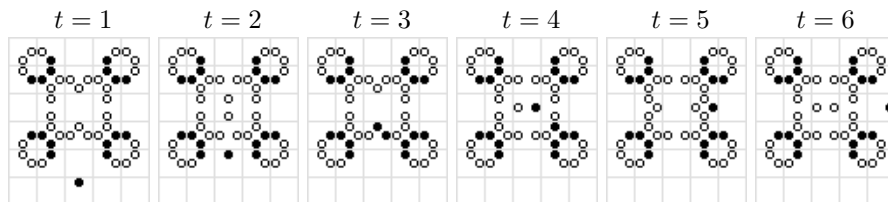
foobar[1]

5 Schluss

Es wurde ein einfaches reversibles Element, das Drehelement, vorgestellt. Mit diesem lassen sich einfach Schaltungen konstruieren. Eine Synchronisierung ist



(a) Der parallele Fall



(b) Der orthogonale Fall

Abbildung 13: Ein Drehelement in P_3 , Screenshots einer Simulation in Golly

dabei nicht notwendig, was das Drehelement vom Fredkin- und Toffoli-Gate unterscheidet. Die Turing-Vollständigkeit solcher Schaltungen wurde demonstriert durch die Konstruktion einer Turingmaschine als Drehelement-Schaltung. Dabei ist die Steuereinheit mit endlicher Anzahl Drehelemente implementierbar. Weiterhin wurde ein reversibler Zellularautomat P_3 vorgestellt, der sehr einfachen Regeln folgt. Es wurde gezeigt, wie sich Drehelemente und deren Schaltungen in P_3 implementieren lassen. Dabei muss lediglich eine gerade Entfernung zwischen allen Drehelementen eingehalten werden. Somit ist die Implementierung sehr einfach möglich.

Literatur

- [1] Kenichi Morita. A simple universal logic element and cellular automata for reversible computing. In *Proceedings of the Third International Conference on Machines, Computations, and Universality*, MCU '01, pages 102–113, London, UK, 2001. Springer-Verlag. ISBN 3-540-42121-1. URL <http://portal.acm.org/citation.cfm?id=648057.744275>.
- [2] C. H. Bennett. Logical reversibility of computation. *IBM J. Res. Dev.*, 17: 525–532, November 1973. ISSN 0018-8646. doi: <http://dx.doi.org/10.1147/rd.176.0525>. URL <http://dx.doi.org/10.1147/rd.176.0525>.
- [3] Kenichi Morita, Akihiko Shirasaki, and Yoshifumi Gono. A 1-tape 2-symbol reversible turing machine. *IEICE Transactions*, E72-E:223–228, March 1989. ISSN 0000-0000.
- [4] Kenichi Morita and Tsuyoshi Ogiro. Embedding a counter machine in a simple reversible 2-d cellular space p_3 . In *Proceedings of the International Workshop on Cellular Automata*, pages 30–31, Osaka, Japan, 2000.

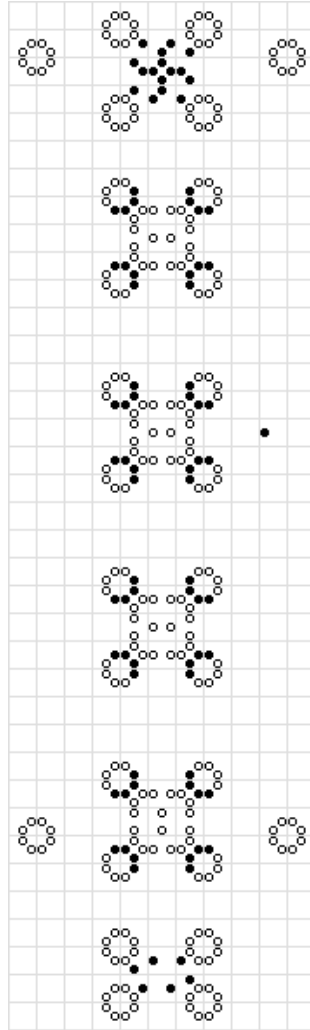


Abbildung 14: Eine Drehelement-Spalte in P_3 , entspricht Abbildung 4

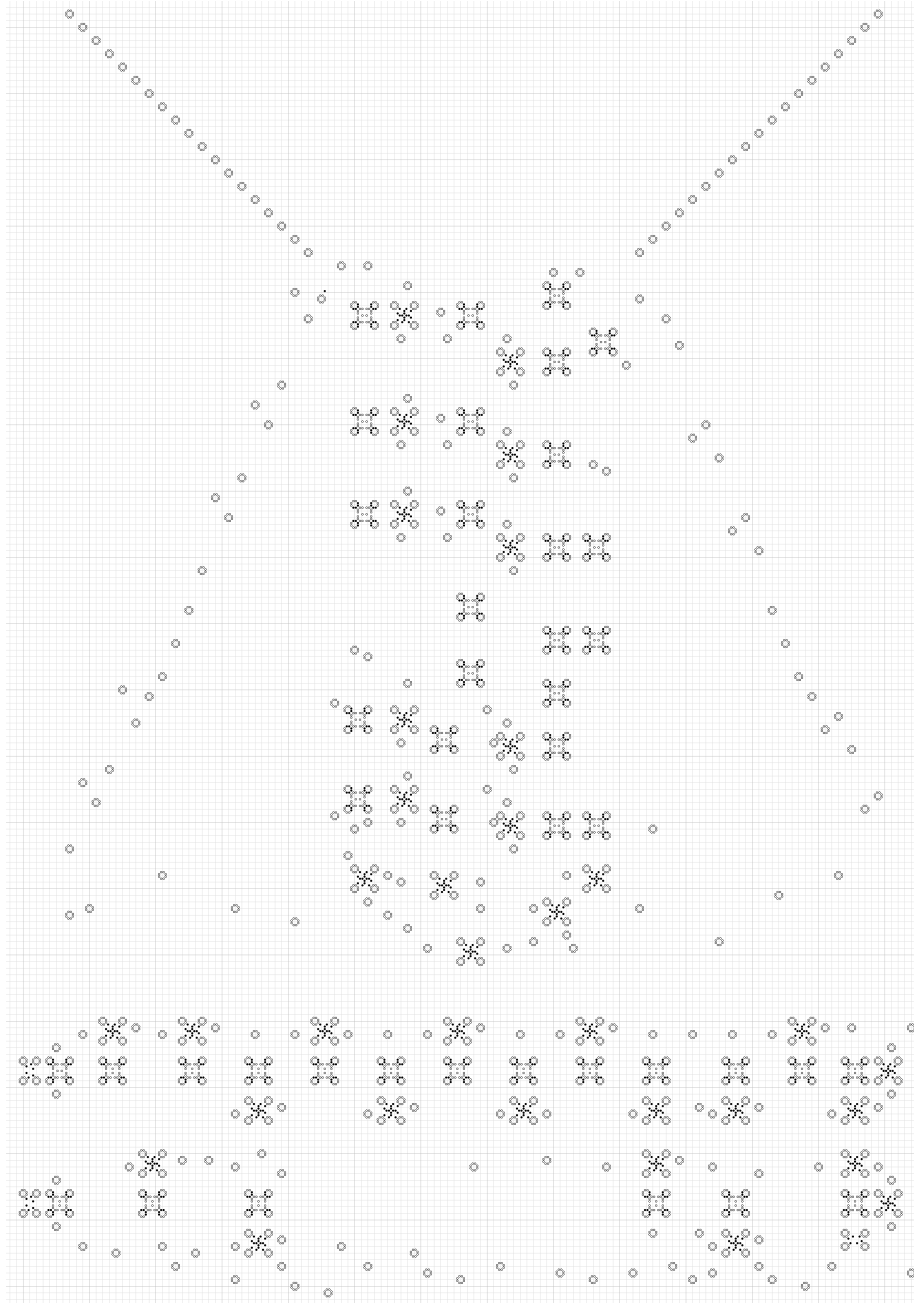


Abbildung 15: Die Turingmaschine T_{2n} in P_3 , entspricht Abbildung 9 (90° gedreht)