

# Ein Einfaches Universales Schaltelement und Zellularautomaten für Rechnerreversibilität

Vortrag von Dennis Felsing  
im Proseminar  
„Zellularautomaten und Diskrete Komplexe Systeme“

Sommersemester 2011



## Zugrunde liegende Quellen



Kenichi Morita.

A Simple Universal Logic Element and Cellular Automata for Reversible Computing.

Proceedings of the Third International Conference on Machines, Computations, and Universality (2001), S. 102-113.



Kenichi Morita.

Embedding a Counter Machine in a Simple Reversible 2-D Cellular Space  $P_3$ .

Proceedings of the International Workshop on Cellular Automata (2000), S. 30-31.



C. H. Bennett.

Logical reversibility of computation.

IBM J. Res. Dev. **17** (1973), S. 525-532.

# Einleitung

- Gewöhnliche Berechnungen nicht reversibel
- Landauer: Wärmeabgabe  $kT \ln 2$  beim Löschen eines Bits
- Ziel: Möglichst einfaches reversibles Element finden und dessen Universalität zeigen  $\Rightarrow$  **Drehelement**
- Außerdem: Als universellen Zellularautomat mit noch einfacheren Regeln aufbauen

## ① Voraussetzungen

- Reversible Mealy-Automaten

- Drehelemente

- Drehelement-Spalten

## ② Reversible Turingmaschinen aus Drehelementen

- Turingmaschinen

- Konstruktion einer Bandzelle

- Konstruktion einer Steuereinheit

- Konstruktion einer reversiblen Turingmaschine

## ③ Reversible Zellularautomaten

- 2-D partitionierter Zellularautomat  $P_3$

- Drehelemente in  $P_3$

# Voraussetzungen

## Überblick

### 1 Voraussetzungen

Reversible Mealy-Automaten

Drehelemente

Drehelement-Spalten

### 2 Reversible Turingmaschinen aus Drehelementen

### 3 Reversible Zellularautomaten

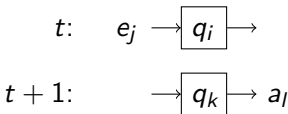
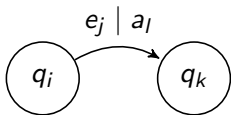
# Reversible Mealy-Automaten (RMA)

$M = (Q, \Sigma, \Gamma, q_1, \delta)$  festgelegt durch

- endliche, nichtleere **Zustandsmenge**  $Q$
- **Eingabealphabet**  $\Sigma$
- **Ausgabealphabet**  $\Gamma$
- Anfangszustand  $q_1 \in Q$  (optional)
- **bijektive Zustandsübergangsfunktion**

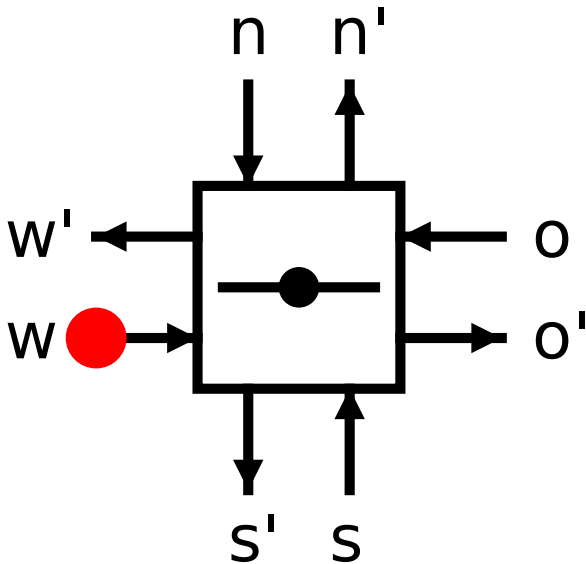
$$\delta : Q \times \Sigma \rightarrow Q \times \Gamma$$

$$\delta(q_i, e_j) = (q_k, a_l)$$



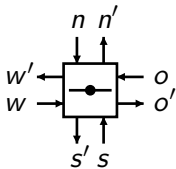
# Drehelemente

Simulation de .res



# Drehelemente

## Formalisierung



Als RMA

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

mit Zustandsübergangsfunktion  $\delta_{DE}$ :

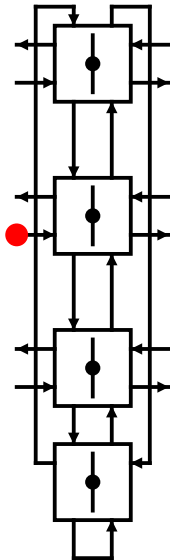
Aktueller Zustand	Eingabe			
	$n$	$o$	$s$	$w$
$\boxed{\bullet \leftarrow}$	$\boxed{\bullet \rightarrow} w'$	$\boxed{\bullet \leftarrow} w'$	$\boxed{\bullet \rightarrow} o'$	$\boxed{\bullet \leftarrow} o'$
$\boxed{\bullet \rightarrow}$	$\boxed{\bullet \rightarrow} s'$	$\boxed{\bullet \leftarrow} n'$	$\boxed{\bullet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

$\delta_{DE}$  bijektiv, also  $M_{DE}$  reversibel



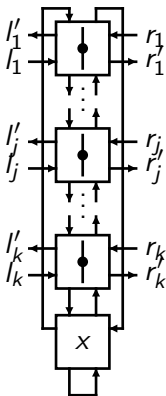
# Drehelement-Spalten

Simulation des .res



# Drehelement-Spalten

## Formalisierung



$$M_{DES} = (\{\boxed{\bullet\rightarrow}, \boxed{\bullet\uparrow}\}, \{l_1, l_2, \dots, l_k, r_1, r_2, \dots, r_k\}, \\ \{l'_1, l'_2, \dots, l'_k, r'_1, r'_2, \dots, r'_k\}, \delta_{DES})$$

Zustandsübergangsfunktion  $\delta_{DES}$ :

Zustand $x$	Eingabe	
	$l_j$	$r_j$
$\boxed{\bullet\rightarrow}$	$\boxed{\bullet\uparrow} l'_j$	$\boxed{\bullet\rightarrow} l'_j$
$\boxed{\bullet\uparrow}$	$\boxed{\bullet\uparrow} r'_j$	$\boxed{\bullet\rightarrow} r'_j$

Realisieren Speicherung eines Bits mit  $k$  „Auslesebahnen“

# Reversible Turingmaschinen aus Drehelementen

Überblick

## ① Voraussetzungen

## ② Reversible Turingmaschinen aus Drehelementen

- Turingmaschinen

- Konstruktion einer Bandzelle

- Konstruktion einer Steuereinheit

- Konstruktion einer reversiblen Turingmaschine

## ③ Reversible Zellularautomaten

# Turingmaschinen

Eine **Einband-Turingmaschine**  $T = (Q, S, q_0, q_f, s_0, \delta)$  ist festgelegt durch

- endliche, nichtleere **Zustandsmenge**  $Q$
- **Bandalphabet**  $S$
- Anfangszustand  $q_0 \in Q$
- Endzustand  $q_f \in Q$
- Blanksymbol  $s_0 \in S$
- partielle **Zustandsübergangsfunktion**

$$\delta \subseteq (Q \times S \times S \times Q) \cup (Q \times \{/\} \times \{-, 0, +\} \times Q)$$

# Turingmaschinen

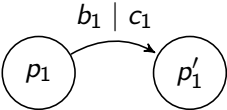
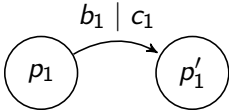
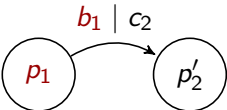
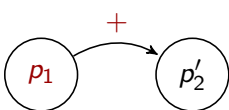
## Determinismus und Reversibilität

Sei  $\alpha_i := [p_i, b_i, c_i, p'_i] \in \delta$ :

**$T$  deterministisch**:  $\Leftrightarrow \neg \exists \alpha_1 \neq \alpha_2 : p_1 = p_2 \wedge (b_1 = b_2 \vee b_2 = /)$

**$T$  reversibel** :  $\Leftrightarrow \neg \exists \alpha_1 \neq \alpha_2 : p'_1 = p'_2 \wedge (c_1 = c_2 \vee b_2 = /)$

Verbotene Übergänge bei deterministischer Turingmaschine:

Quadrupel	1. Fall	2. Fall
$\alpha_1$		
$\alpha_2$		

# Turingmaschinen

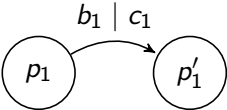
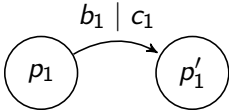
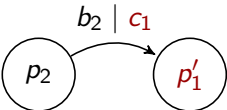
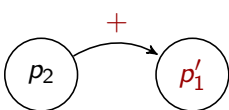
## Determinismus und Reversibilität

Sei  $\alpha_i := [p_i, b_i, c_i, p'_i] \in \delta$ :

**$T$  deterministisch**:  $\Leftrightarrow \neg \exists \alpha_1 \neq \alpha_2 : p_1 = p_2 \wedge (b_1 = b_2 \vee b_2 = /)$

**$T$  reversibel** :  $\Leftrightarrow \neg \exists \alpha_1 \neq \alpha_2 : p'_1 = p'_2 \wedge (c_1 = c_2 \vee b_2 = /)$

Verbotene Übergänge bei reversibler Turingmaschine:

Quadrupel	1. Fall	2. Fall
$\alpha_1$		
$\alpha_2$		

# Turingmaschinen

## Universalität und Konstruktion

### Theorem

*Für jede Einband-Turingmaschine gibt es eine reversible (semiunendliche) Einband-Zwei-Symbol-Turingmaschine, welche die Erstere simuliert.*

### Konstruktion

Bestandteile einer reversiblen Turingmaschine:

- Semiunendliches Band bestehend aus **Bandzellen**
- **Steuereinheit**: Implementiert Zustandsübergangsfunktion  $\delta$

# Konstruktion einer Bandzelle

Eingänge

Befehle von Steuereinheit an Bandzelle an Kopfposition:

$\mathbb{B} := \{ \text{R}, \text{W}, \text{SL}, \text{SR}, \text{E0}, \text{E1} \}$

Signale zwischen benachbarten Bandzellen (intermediate):

$\mathbb{I} := \{ \text{SLI}, \text{SRI} \}$

Antworten von Bandzelle an Steuereinheit (completion):

$\mathbb{C} := \{ \text{Rc0}, \text{Rc1}, \text{Wc}, \text{SLc}, \text{SRc}, \text{Ec} \}$

Lesen    Schreiben    Kopfbewegung    Löschen



# Konstruktion einer Bandzelle

## Formalisierung

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

$$Q_{BZ} = \{(k, z) \mid k, z \in \{0, 1\}\}$$

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

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

# Konstruktion einer Bandzelle

Formalisierung: Zustandsübergänge

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

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

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

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

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

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

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

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

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

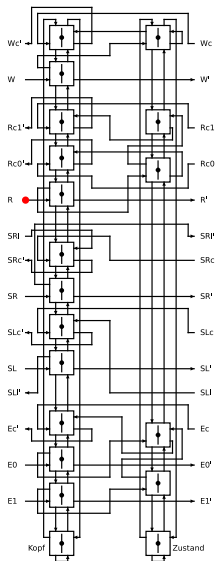
$$\delta_{BZ}((1, z), SL) = (0, z, SLI') \quad (9)$$

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

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

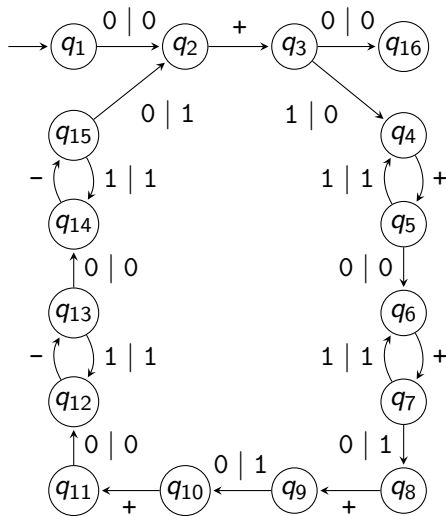
# Konstruktion einer Bandzelle

Simulation bzm.res



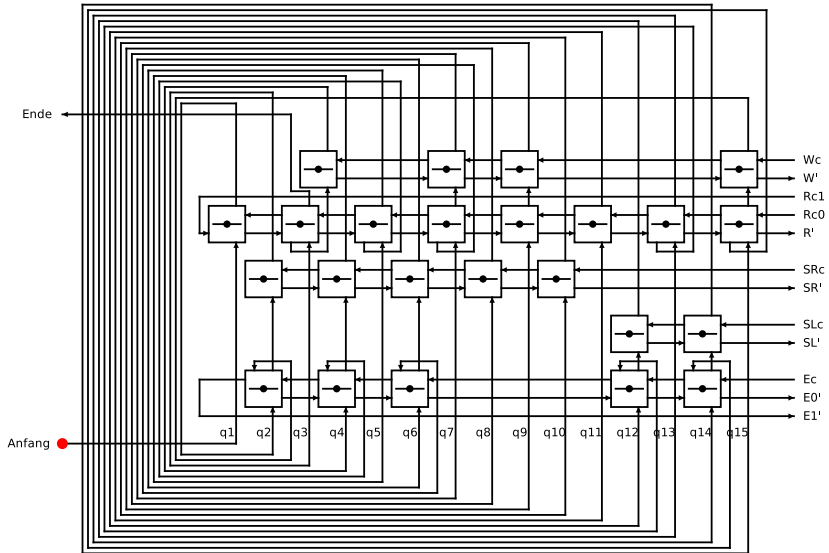
## Konstruktion einer Steuereinheit

$T_{2n}$  berechnet  $f(n) = 2n$  für unär kodierte  $n$  auf Band:



# Konstruktion einer Steuereinheit

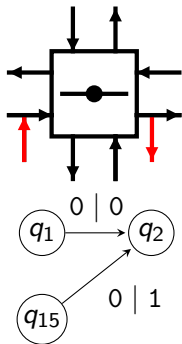
Simulation steuereinheit.res



# Konstruktion einer Steuereinheit

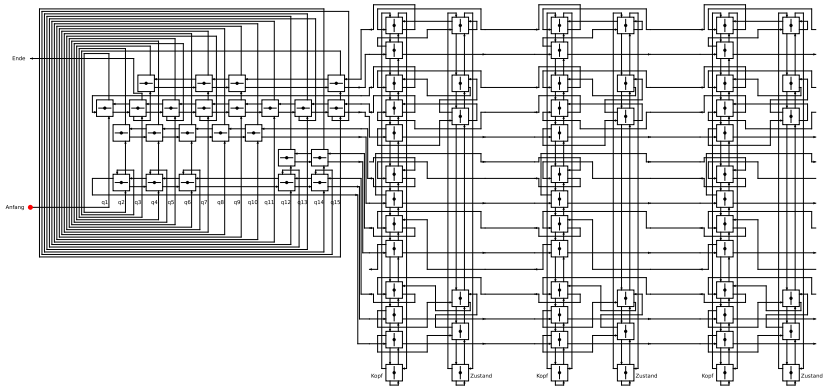
## Reversibles Löschen

- Löschen inverse Operation zum Lesen
- Verbindungen in Drehelement-Schaltung dürfen nicht auseinander- oder zusammenlaufen  
⇒ Zwei Übergänge in einen Zustand nicht in Steuereinheit möglich
- Lösung: Vergessen ob 0 oder 1 gelesen wurde  
Dazu Bandzelle gelesenen Wert übergeben



# Konstruktion einer reversiblen Turingmaschine

Simulation t2n-3.res





# Reversible Zellularautomaten

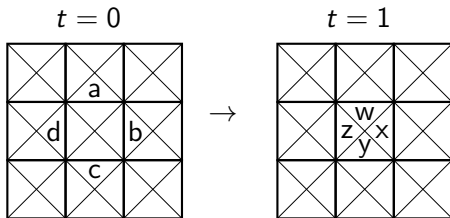
## Überblick

- 1 Voraussetzungen
- 2 Reversible Turingmaschinen aus Drehelementen
- 3 Reversible Zellularautomaten
  - 2-D partitionierter Zellularautomat  $P_3$
  - Drehelemente in  $P_3$



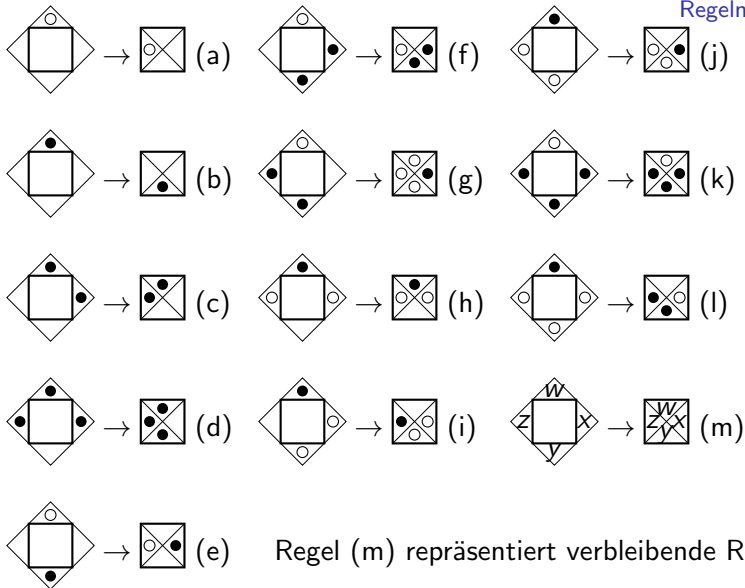
## 2-D partitionierter Zellularautomat $P_3$

- Jede Zelle in 4 Teile partitioniert: 
- Jeder Teil kann 3 Zustände annehmen: 
- Zustand einer Zelle von benachbarten Teilen abhängig
- Alle Regeln **rotationssymmetrisch**
- **Reversibel**, da es keine zwei verschiedenen Regeln gibt, die Feld in gleichen Zustand bringen



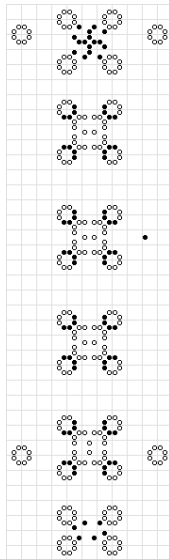
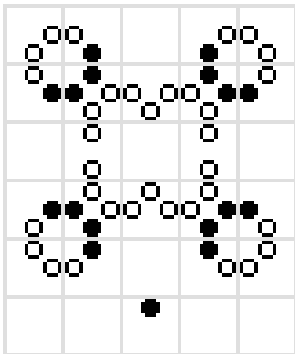
## 2-D partitionierter Zellularautomat

Regeln



## Drehelemente in $P_3$

Simulationen de.rle, des.rle, t2n-3.rle



- Sehr einfaches logisches Element: **Drehelement**
- Turingmaschine lässt sich damit einfach konstruieren:
  - Endliche Schaltung für Steuereinheit
  - Keine Synchronisierung notwendig
- Drehelemente und -schaltungen lassen sich in einfachen reversiblen Zellularautomaten  $P_3$  implementieren