Ein Einfaches Univerales Schaltelement und Zellularautomaten für Rechnerreversibilität

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

Sommersemester 2011

Karlsruhe Institute of Technology

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

Überblick

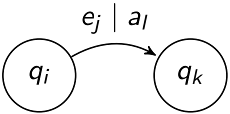
1. Voraussetzungen
   - Reversible Mealy-Automaten
   - Drehelemente
   - Drehelement-Spalten
2. Reversible Turingmaschinen aus Drehelementen
   - Turingmaschinen
   - Konstruktion einer Bandzelle
   - Konstruktion einer Steuereinheit
   - Konstruktion einer reversiblen Turingmaschine
3. Reversible Zellularautomaten
   - 2-D partitionierter Zellularautomat P_3
   - Drehelemente in P_3

Reversible Mealy-Automaten (RMA)

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

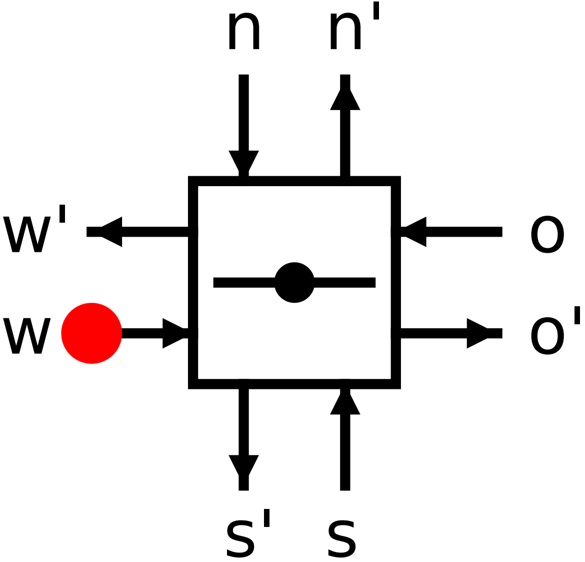
- endliche, nichtleere Zustandsmenge Q
- Eingabealphabet Σ
- Ausgabealphabet Γ
- Anfangszustand $q_1 \in Q$ (optional)
- **bijektive** Zustandsübergangsfunktion
  $\delta : Q \times \Sigma \to Q \times \Gamma$

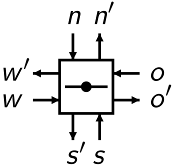
$$\delta(q_i, e_j) = (q_k, a_l)$$



t: $\quad e_j \longrightarrow \boxed{q_i} \longrightarrow$

$t+1$: $\quad \longrightarrow \boxed{q_k} \longrightarrow a_l$


n
n'
w'
o
w
o'
s'
s



Als RMA

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

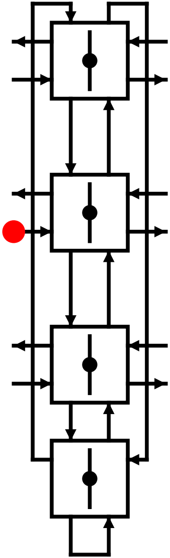
mit Zustandsübergangsfunktion δ_{DE}:

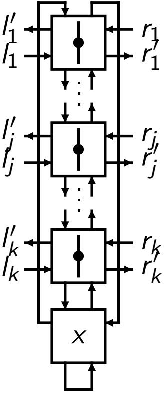
Aktueller Zustand	Eingabe n	Eingabe o	Eingabe s	Eingabe w
$\boxed{-}$	$\boxed{\vert}$ w'	$\boxed{-}$ w'	$\boxed{\vert}$ o'	$\boxed{-}$ o'
$\boxed{\vert}$	$\boxed{\vert}$ s'	$\boxed{-}$ n'	$\boxed{\vert}$ n'	$\boxed{-}$ s'

δ_{DE} bijektiv, also M_{DE} reversibel

Drehelement-Spalten

Simulation `des.res`





$$M_{DES} = (\{\boxed{-}, \boxed{|}\}, \{l_1, l_2, \ldots, l_k, r_1, r_2, \ldots, r_k\},$$
$$\{l'_1, l'_2, \ldots, l'_k, r'_1, r'_2, \ldots, r'_k\}, \delta_{DES})$$

Zustandsübergangsfunktion δ_{DES}:

	Eingabe	
Zustand x	l_j	r_j
$\boxed{-}$	$\boxed{\vert}$ l'_j	$\boxed{-}$ l'_j
$\boxed{\vert}$	$\boxed{\vert}$ r'_j	$\boxed{-}$ r'_j

Realisieren Speicherung eines Bits mit k „Auslesebahnen“

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

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

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

$$T\,\textbf{reversibel} :\Leftrightarrow \neg\exists \alpha_1 \neq \alpha_2 : p_1' = p_2' \wedge (c_1 = c_2 \vee b_2 = /)$$

Verbotene Übergänge bei deterministischer Turingmaschine:

Quadrupel	1. Fall	2. Fall
α_1	$p_1 \xrightarrow{b_1 \mid c_1} p_1'$	$p_1 \xrightarrow{b_1 \mid c_1} p_1'$
α_2	$p_1 \xrightarrow{b_1 \mid c_2} p_2'$	$p_1 \xrightarrow{+} p_2'$

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
α_1	$p_1 \xrightarrow{b_1 \mid c_1} p'_1$	$p_1 \xrightarrow{b_1 \mid c_1} p'_1$
α_2	$p_2 \xrightarrow{b_2 \mid c_1} p'_1$	$p_2 \xrightarrow{+} p'_1$

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 δ

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

$$
\begin{aligned}
M_{BZ} &= (Q_{BZ}, \Sigma_{BZ}, \Gamma_{BZ}, \delta_{BZ}) \\
Q_{BZ} &= \{(k, z) \mid k, z \in \{0, 1\}\} \\
\Sigma_{BZ} &= \mathbb{B} \cup \mathbb{I} \cup \mathbb{C} \\
\Gamma_{BZ} &= \{x' \mid x \in \Sigma_{BZ}\}
\end{aligned}
$$

Konstruktion einer Bandzelle

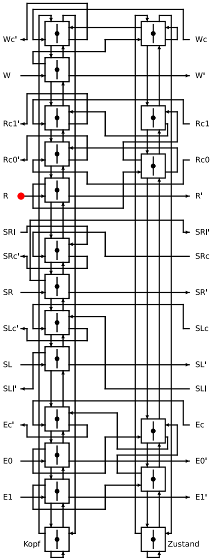
Formalisierung: Zustandsübergänge

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

$$
\begin{aligned}
\delta_{BZ}((0, z), y) &= (0, z, y') && (1)\\
\delta_{BZ}((0, z), SRI) &= (1, z, SRc') && (2)\\
\delta_{BZ}((0, z), SLI) &= (1, z, SLc') && (3)\\
\delta_{BZ}((1, 0), R) &= (1, 0, Rc0') && (4)\\
\delta_{BZ}((1, 1), R) &= (1, 1, Rc1') && (5)\\
\delta_{BZ}((1, 0), W) &= (1, 1, Wc') && (6)\\
\delta_{BZ}((1, 1), W) &= (1, 0, Wc') && (7)\\
\delta_{BZ}((1, z), SR) &= (0, z, SRI') && (8)\\
\delta_{BZ}((1, z), SL) &= (0, z, SLI') && (9)\\
\delta_{BZ}((1, 0), E0) &= (1, 0, Ec') && (10)\\
\delta_{BZ}((1, 1), E1) &= (1, 1, Ec') && (11)
\end{aligned}
$$

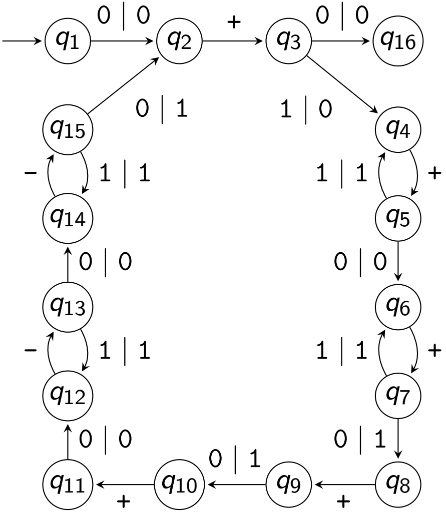
Konstruktion einer Bandzelle

Simulation `bzm.res`



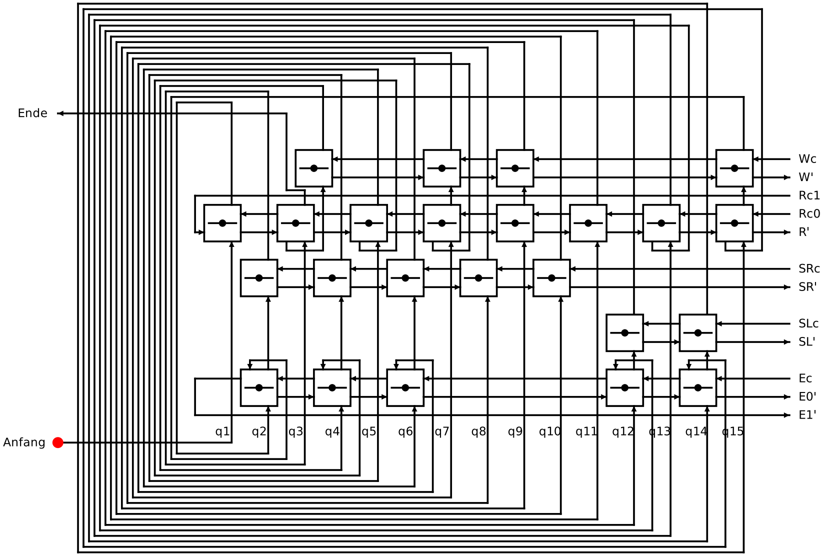
Konstruktion einer Steuereinheit

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



Konstruktion einer Steuereinheit

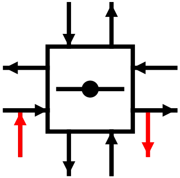
Simulation `steuereinheit.res`

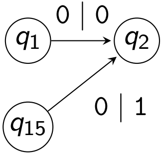


Konstruktion einer Steuereinheit

Reversibles Löschen

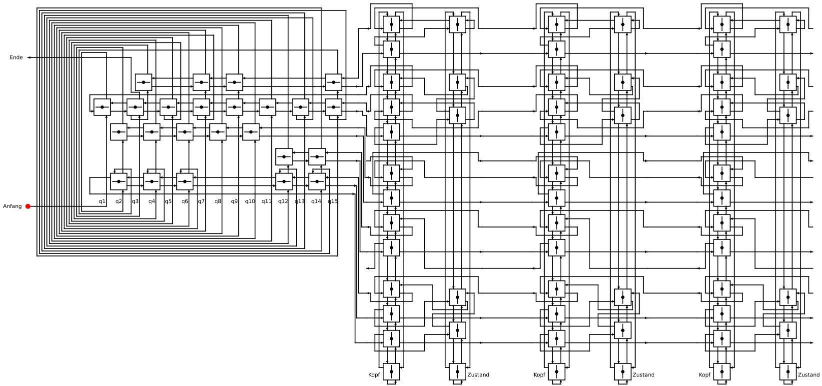
- Löschen inverse Operation zum Lesen
- Verbindungen in Drehelement-Schaltung dürfen nicht auseinander- oder zusammenlaufen
  $\Rightarrow$ Zwei Übergänge in einen Zustand nicht in Steuereinheit möglich
- Lösung: Vergessen ob 0 oder 1 gelesen wurde
  Dazu Bandzelle gelesenen Wert übergeben





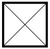
Konstruktion einer reversiblen Turingmaschine

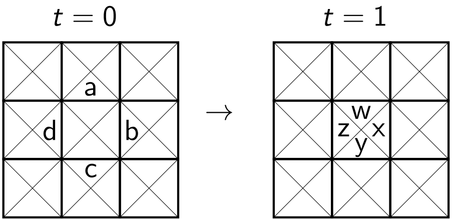
Simulation `t2n-3.res`



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

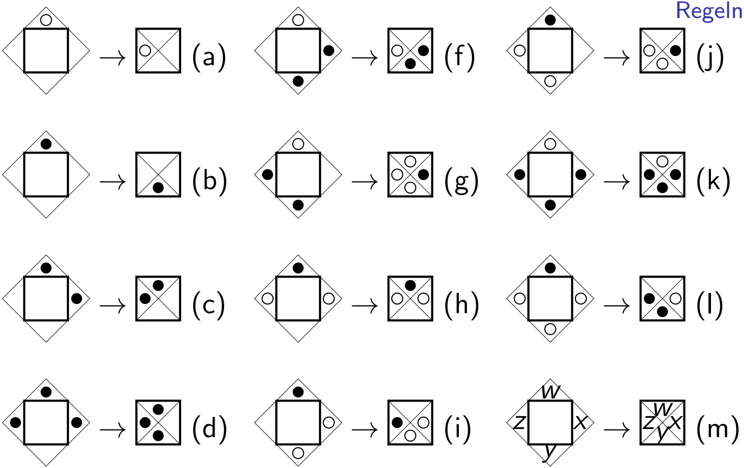
2-D partitionierter Zellularautomat P_3

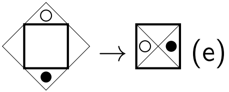
- Jede Zelle in 4 Teile partitioniert: 
- Jeder Teil kann 3 Zustände annehmen: , , 
- Zustand einer Zelle von benachbarten Teilen abhängig
- Alle Regeln rotationssymetrisch
- **Reversibel**, da es keine zwei verschiedenen Regeln gibt, die Feld in gleichen Zustand bringen



2-D partitionierter Zellularautomat

Regeln

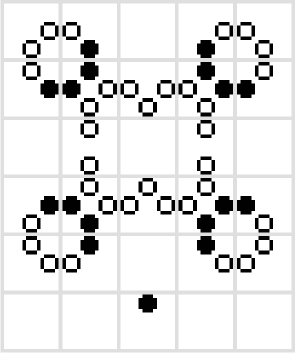


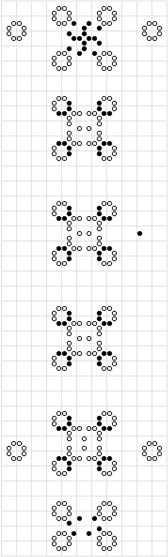


Regel (m) repräsentiert verbleibende Regeln

Drehelemente in P_3

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





Abschluss

- 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